

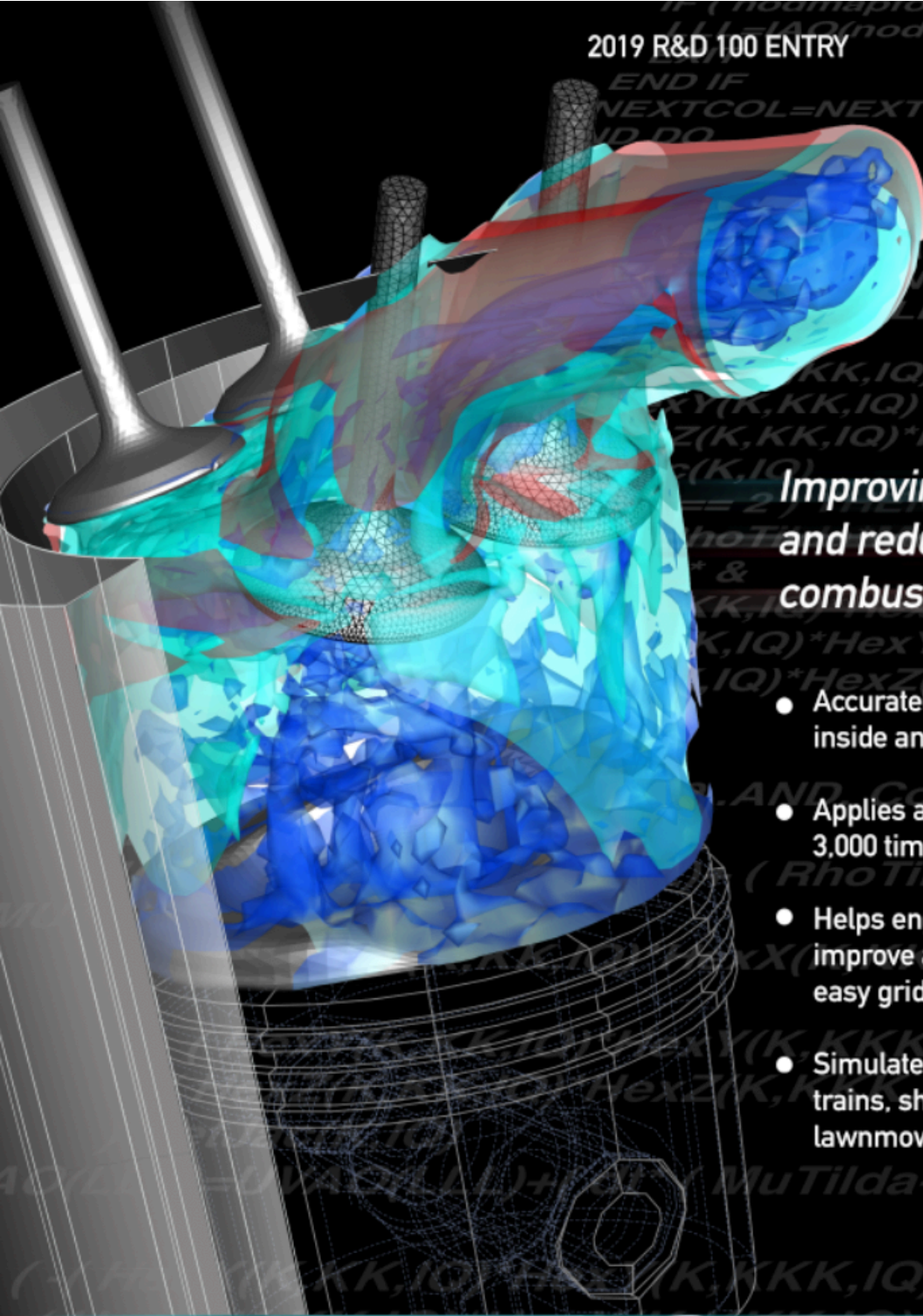
LA-UR-19-23015

Approved for public release; distribution is unlimited.

Title:	FEARCE: Fast, Easy, Accurate, and Robust Continuum Engineering Improving fuel efficiency and reducing emissions in combustion engines 2019 R&D 100 Awards
Author(s):	Carrington, David Bradley Ramos, Octavio Jr
Intended for:	2019 R&D 100 Awards submission Report
Issued:	2019-04-17 (rev.1)

Disclaimer:

Los Alamos National Laboratory, an affirmative action/equal opportunity employer, is operated by Triad National Security, LLC for the National Nuclear Security Administration of U.S. Department of Energy under contract 89233218CNA000001. By approving this article, the publisher recognizes that the U.S. Government retains nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or to allow others to do so, for U.S. Government purposes. Los Alamos National Laboratory requests that the publisher identify this article as work performed under the auspices of the U.S. Department of Energy. Los Alamos National Laboratory strongly supports academic freedom and a researcher's right to publish; as an institution, however, the Laboratory does not endorse the viewpoint of a publication or guarantee its technical correctness.



Improving fuel efficiency and reducing emissions in combustion engines

- Accurately simulates turbulence and mixing inside an engine
- Applies a revolutionary parallel code about 3,000 times faster than today's serial codes
- Helps engine designers and engineers improve and optimize engines by quick and easy grid generation
- Simulates any motor used in vehicles, aircraft, trains, ships, turbines, and your everyday lawnmower or boiler

FEARCE

Fast, Easy, Accurate, and Robust Continuum Engineering

David B. Carrington and Jiajia Waters

2019 R&D 100 Awards template

FEARCE: Fast, Easy, Accurate, and Robust Continuum Engineering

Improving fuel efficiency and reducing emissions in combustion engines

LA-UR-19-

Categories

- | | |
|----------------------------------------------------------|--------------------------------------------------------------------------------------|
| <input type="checkbox"/> Analytical/Test | <input type="checkbox"/> Special recognition: Corporate Social Responsibility |
| <input type="checkbox"/> IT/Electrical | <input checked="" type="checkbox"/> Special recognition: Green Tech |
| <input checked="" type="checkbox"/> Mechanical/Materials | <input checked="" type="checkbox"/> Special recognition: Market Disruptor - Products |
| <input type="checkbox"/> Process/Prototyping | <input type="checkbox"/> Special recognition: Market Disruptor - Services |
| <input checked="" type="checkbox"/> Software/Services | |
| <input type="checkbox"/> Other | |

Note: There is a \$450 entry fee per category

Name of primary submitting organization

Los Alamos National Laboratory

Name(s) of co-developing organization(s)—if applicable

N/A

Product/service brand name

FEARCE

Was the product/service introduced to the market between January 1, 2018, and March 31, 2019?

- ☒ Yes
☐ No

If your submission is subject to regulatory approval, has the product been approved?

- ☐ Yes
☐ No
☒ Not applicable to this product

Price of product/service (U.S. dollars)

As business plans are being formalized, the price for FEARCE is anticipated to be less than the amounts listed below. Please feel free to contact the Los Alamos National Laboratory Richard P. Feynman Center for Innovation for further information.

Anticipated Use	Anticipated Cost
University Research/Academic	> \$10,000
Standard Commercial Use	> \$100,000

Product description

Short for Fast, Easy, Accurate, and Robust Continuum Engineering, FEARCE is a new software system that predicts complex turbulent flows in combustion engines. Such predictions help designers improve engine performance, enhance fuel efficiency, and reduce pollutant emissions. When it comes to improving engine performance, it doesn't get any more FEARCE.

Indicate the type of institution you represent

Government Laboratory

Submitter's relation to entered product/service

Product Developer

Product Photos

2019 FEARCE R&D 100 Cover
FEARCE 1
FEARCE 2

Video Files

2019 FEARCE Video

What does the product or technology do? Describe the principal applications of this product.

In the United States alone, more than 250 million vehicles rely on the tried-and-true internal combustion engine. As gasoline and diesel become more expensive and alternative fuels become more available, vehicle manufacturers around the world are investing time and effort into improving these familiar motors. (For an overview of such efforts, read the article “*Don’t Forget the Combustion Engine*” in the Appendix.)

Improving the efficiency of internal combustion engines is one of the most promising and cost-effective near- to mid-term approaches to increasing engine power while improving a vehicle’s fuel economy and reducing toxic emissions that pollute the environment.

To improve the internal combustion engine, engine designers are particularly interested in turbulence—the swirling, violent chaos that results from mixing fuel with gases—when fuel burns. By better understanding and predicting the effects of turbulence on an engine’s energy efficiency, researchers can anticipate and manipulate fluid dynamics to modify and thereby improve an engine’s performance.

The only real way to test how experimental components might improve mixing is to build an experimental engine. But physically modifying and testing parts experimentally to determine if the modifications will actually improve engine performance is very time-consuming and can cost many millions of dollars.

In an innovative, time- and cost-saving solution to this problem, scientists at Los Alamos National Laboratory have developed a multiphysics software system known as FEARCE (Fast, Easy, Accurate, and Robust Continuum Engineering) that allows researchers to invent, modify, and test parts from the comfort of a desktop computer or even on a supercomputer.

FEARCE investigates and simulates turbulent chemically reactive flows in fuel combustion, such as fuel injection and fuel-air mixing, thermodynamic combustion losses, and combustion–emission formation processes in engines. Our objective is to understand turbulent flows and how they react and interact with other complicating variables, including highly nonlinear heat and mass transfer, small-scale velocity, and chemical kinetics or reactions. Our emphasis is on the fluid dynamics and chemical reactions that take place in all kinds of internal combustion engines, including the

reactions between multiple species (such as air, gasoline, carbon dioxide, water, and hundreds of other species) and multiphase (liquid and gas) fluid flow.

Our principal concern is with flow associated to engines, particularly internal combustion engines, such as gasoline and diesel engines that power automobiles and trucks. However, FEARCE can be used to simulate any type of engine, such as large-bore engines found in ships and trains, as well as turbines, gas burners, heaters, boilers, and your everyday lawnmowers.

FEARCE models an engine's operating properties and ranges that can't be addressed with experiments. These simulation tools enable us to benchmark areas of most interest for engine operation before experiments are conducted. We also use these software tools to understand or analyze experimental data from our collaborators.

Engine designers can use the FEARCE simulations to optimize existing engine designs for increased fuel efficiency and reduced emissions and to understand how any changes affect the efficiency and operation of engine. FEARCE also models new engine components and runs simulations that show how such components influence and perhaps improve the mixing process, thus changing the turbulence inside an engine to generate more power with less fuel while releasing fewer pollutants.

Using FEARCE also enables designers to develop and optimize engines to run on alternative fuels, such as biofuels, which may require different operating conditions than those required for conventional fuels.

FEARCE models five key components of an internal combustion engine's operation, described below:

- **Modeling the Combustion Process**

Modern-day engines have sophisticated geometries in which liquid fuel, such as gasoline, is injected and atomized into small drops. These small droplets evaporate into gaseous fuel, making it available for a reaction known as combustion. FEARCE simulates the combustion process by using a robust and high-fidelity chemical kinetics reaction model.

- **Simulating the Engine's Fuel Spray Injection Process**

FEARCE has a choice when it comes to simulating the spray injection process: Taylor-Analogy Breakup (TAB), the Kelvin Helmholtz–Rayleigh Taylor model (KH-RT), and the Volume of Fluids (VOF) method. TAB and KH-RT use a stochastic Lagrangian Particle System. This system tracks, diffuses, and applies turbulent dispersion to parcels (droplets). These droplets represent many particles with properties distributed about a mean. The droplets also affect a fluid's motion as they pass through. In Figure 1, the KH-RT model is shown with the droplets transported in the gases by a Lagrangian particle method, where the drops and gas interact with each other, including heat and mass transfer by evaporation of drops.

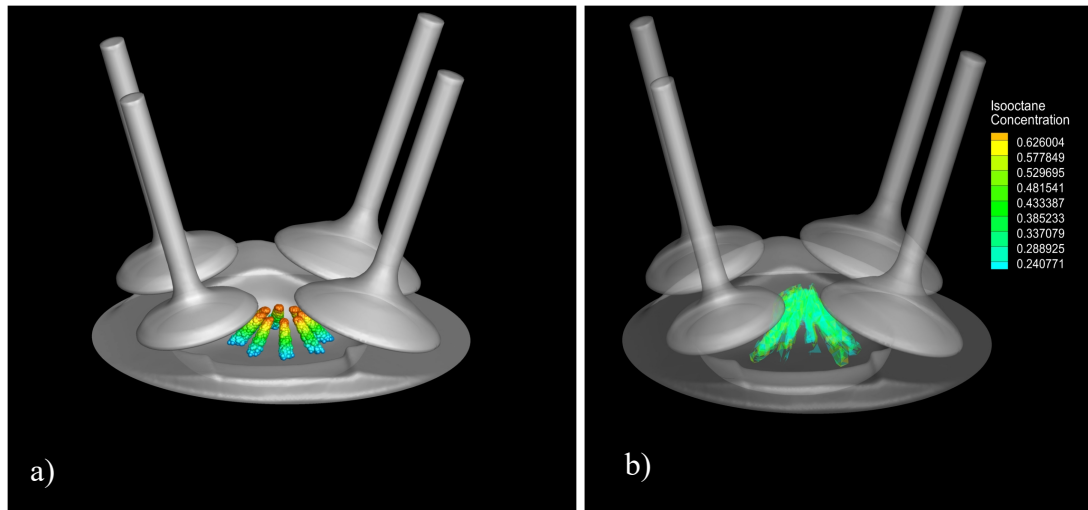


Figure 1. Gasoline injected into 4-valve direct injection spark injection engine of modern design. In this view, the four valves are shown on top, with the colored fuel spray, and the piston below: a) Liquid fuel injected using the KH-RT spray break-up modeling combined with b) droplet evaporation model showing gaseous fuel.

- **Simulating Particle Transport and Fluid-Droplet Force Balances**

Unlike any other software system, FEARCE uses what it called a Finite-Element Method (FEM) system to evaluate the fluid and droplet properties at the location in which they reside. The fluid is treated as a continuum, one that varies continuously within each small element that represents the fluid in that portion of the problem. Other software systems rely on nearest node or element property, thus not truly representing fluids as continuous but rather as merely piecewise or element-wise averages. With FEARCE, it is possible to simulate more accurate particle transport and fluid-droplet force balances.

- **Modeling the Spray Injection Process, Fuel Burn, and How Vaporized Gases Supply Heat and Create Pressure onto an Engine's Pistons**

To model the spray injection process and the liquid jet's initial breakup into ligaments, FEARCE uses a Volume of Fluids (VOF) method. Simply put, spray injection is the breaking up of the liquid jet into large drops. The VOF method in FEARCE calculates the exact forces on the liquid jet and large drops to break them into smaller droplets.

FEARCE also uses a LANL-developed droplet vaporization modeling process, which simulates fuel burn using the chemical kinetics portion of the software.

FEARCE simulates how the vaporized gases burn or combust, supplying heat and creating high pressure onto the engine's pistons. The expanding gases and high pressure from the combustion process move the pistons. FEARCE captures the thermodynamics and fluid dynamic processes, including the turbulence of the fluid and fuel mixtures as they move throughout the engine system.

- **Modeling How Moving Parts Function in an Engine**

In an engine, pistons compress the fluid to high pressures and temperatures before the fuel is ignited. Valves in the engine open and close, thus controlling the inflow and outflow of the fluids while also promoting mixing. Pistons also extract energy released from the burning fuels as higher pressure drives the piston downward, expanding the volume and converting the energy released into work.

To simulate these processes, FEARCE uses a sophisticated calculus known as the immersed boundary method (IBM) or immersed finite-element method (IFEM). FEARCE uses one of two different methodologies for this calculus: (1) interpolating polynomials designed to evaluate the current state of the fluid in the neighborhood of the moving parts (IBM) or (2) special polynomials and a fictitious fluid method, the latter known as the IFEM method. This latter method is more accurate, but it comes with the extra cost of evaluating the entire domain, instead of just those fluid elements that are currently part of the engine processes (i.e., those above the piston or active cells). These two options are the first of their kind in the engine-modeling

community, which until now has relied on cut-cell systems and the Arbitrary Lagrangian Eulerian (ALE) method.

Cut-cell systems hack out of a regular orthogonal grid (a small rectangular box) the engine geometry and the moving parts, creating polyhedrals at the boundaries of these surfaces, often producing irregular shapes. The scheme suffers from difficulty in maintaining high accuracy at the boundaries and is not extensible to higher order accuracy produced by the FEM systems.

The ALE method was designed to achieve good accuracy of a fluid's advection (the transport of matter by the flow of the fluid) because the grid moves with the fluid during the advection process. Diffusion processes are performed on the Eulerian grid or diced on the moved grid. To avoid the grid from becoming a tangled, incalculable mess, it must be moved back to some reasonable shape, thus requiring a remapping step. Often, the remapping fluxes (passes) the mass back through the grid as the grid moves back to near its original shape before the next increments in time are simulated. Such fluxing also causes error because there is an advection process that has significant numerically related diffusion.

We developed FEARCE to address and overcome these difficulties, which are associated with the mainstream engine-modeling community, while also providing faster and easier grid generation, a more accurate and robust solution, and an increase speed of solution.

FEARCE's multiphysics software delivers multi-species modeling and simulation capabilities for complex, turbulent, and reactive flows. FEARCE is the first software system that designers and engineers can use to better understand the nature of turbulence and how it affects the efficiency of fuel mixing with gases to create optimum energy, which translates to more efficient vehicle power. FEARCE can also model new engine components and run simulations that show how new components influence and improve the mixing process, thus changing the turbulence inside an engine so that it generates more power with less fuel while producing fewer pollutants into the air. FEARCE can also optimize engine designs for alternative fuels, adapting engines to the alternative

fuels' different operating conditions. Cost savings in engine design come from using the software to develop an optimum design that only needs to be physically built once.

For an overview of FEARCE and its key features, read the “Request for Information for FEARCE: An Extensible Finite Element Multiphysics Software Suite for Complex and Reactive Modeling,” as well as “2018 FEARCE Development: A Robust and Accurate Modeling Software,” both in the Appendix.

The following support letters (see Appendix) address FEARCE’s features and applications:

- Michael R. Weismiller, U.S. Department of Energy, Vehicle Technologies Office
- Juan C. Heinrich, University of New Mexico School of Engineering, Department of Mechanical Engineering
- Peter R. Eiseman, GridPro, Program Development Company
- Peng Zhao, Oakland University, Department of Mechanical Engineering
- Darrell W. Pepper, University of Nevada Las Vegas, Department of Mechanical Engineering
- Haiwen Ge, Texas Tech University, Department of Mechanical Engineering

How does the product operate? Describe the mechanism of action, theories, materials, composition, or construction.

A multiphysics software system, FEARCE operates on any type of computer, from a laptop to the world's fastest supercomputer. All that matters is for a user to determine the complexity of the problem and establish the desired accuracy of the solution.

The theory is based on the first law of thermodynamics: the conservation of mass, momentum, and energy. These physical quantities are simulated by mathematical approximation of the equations that represent the conservation of these properties as derived over a volume or cell—this is what acts on the cell's surface, goes through the surfaces, and is generated within the volume to change the mass, momentum, and energy of the fluid within the element.

FEARCE approximates a large system by using thousands to tens of thousands to even millions of small cells or elements to represent small pieces of the larger system, such as an engine. In other words, FEARCE dissects an engine into small, easily handled pieces.

FEARCE solves the equations related to each small piece, keeping in mind how each piece is related to every other small piece in the “neighborhood.” FEARCE then integrates the small pieces, the resultant evaluation creating a picture of the effect of conservation laws and external forces applied to the system, such as the motion of an engine's piston.

FEARCE conserves the physical properties of each small domain—in entirety, these small domains represent the whole domain. FEARCE can change these properties throughout the machine, in effect connecting the smaller elements (represented in FEARCE by hexahedrals) and the element nodes (vertexes).

FEARCE approximates these conservation equations on each little element, with each approximation as accurate as the level of approximation and level of resolution (i.e., how small the cells are). Overall, the simulation of the behavior or the physics in these machines is accurate to a desired percentage, which is often very small. This percentage is partly related to the size of the small elements and also the order or the polynomial making the approximation of the equations on each element. This approximation method in FEARCE is known as FEM.

FEM offers three key advantages over other simulation or modeling systems:

- Provides enhanced scalability of the software on parallel machines,
- Makes it easy to perform higher accuracy simulations, and
- Facilitates what is called grid generation, the creation of simulated parts or engines.

To run efficiently on all kinds of computers, millions of cells require many processing elements of computational units, known as Central Processing Units (CPUs). A large domain is split up into these CPUs for solution, with thousands of elements going to each CPU, where they are solved. Each CPU's solution is communicated to other CPUs for aggregation of the whole—this is done through an MPI, a Message Passing Interface.

FEARCE achieves this process by applying FEM, making the domain splitting (see Figure 2) efficient so that it is better than the doubling idea (sometimes referred to as ideal scaling)—in which a doubling in the problem size, along with the number of processors, doubles the time it takes to solve (this is called naïve or weak scaling).

Because FEARCE is superlinear (a function that eventually grows faster than any linear one), it falls well below the linear curve and in fact has an almost flat response when using sophisticated equation solution systems like multi-grid methods. This superlinear capability or high scalability comes from the unique way in which FEARCE's integration algorithm constructs and performs the differential operators in a localized manner, removing extensive communication costs and operator construct across domains. Integration is an additive process (every element's integration is done locally), and what's missing on nodes that share a domain is simply added to what was calculated from other domains at the end of the integration process. This is just one example of the advantages FEARCE has over its competitors, none of which can generally develop superlinear behavior (see Figure 2).

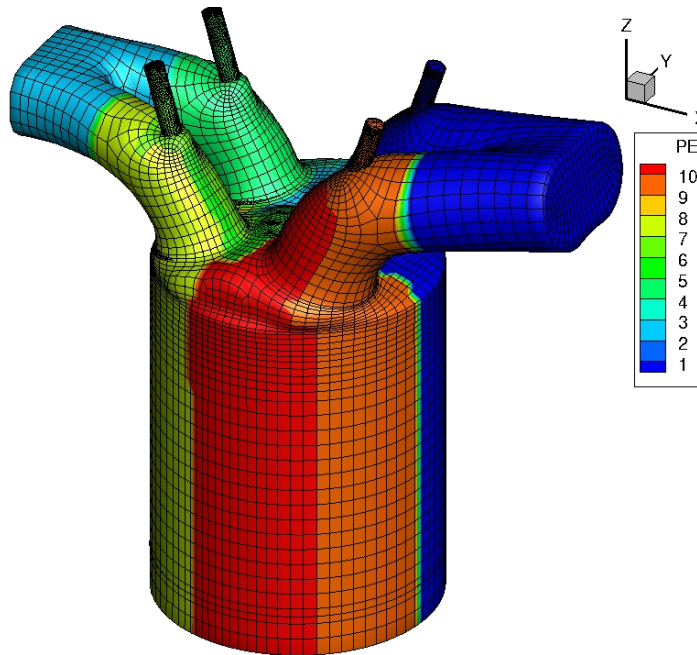


Figure 2. The many domains of the engine's combustion chamber and ports (10 domains). Each domain is handled by a CPU. The way solution process is done with FEARCE, these domains only need to communicate at the end of the integration process forming the whole, making FEARCE highly scalable.

Although the representation of the physics, the domain, and the moving parts on the computer are all binary, FEARCE itself is written in a computer language (Fortran 95). This language makes it possible for a compiler to interpret and convert it to binary instruction sets, or machine language. FEARCE is written in a modular way, with the FEM workings deep within the code, where people don't have to work. It thus is the machinery of approximation that requires no adjustment, as models are changed or added at the code's higher level. The code is therefore considered to be levelized and modular, both desirable features of a modern-day software.

A rather unique method to the engine-modeling community is how FEARCE handles moving parts robustly and accurately, never failing during simulation. The unique immersed moving parts methods facilitates the ease of grid generation. FEARCE includes high-fidelity chemical reaction kinetics to enable evaluation of emissions providing for development of cleaner, more efficient engine design and the use of new fuels, such as biofuels or even multiple fuels at once.

The following papers in the Appendix provide additional details about FEARCE:

- “Turbulent reactive flow modeling in engines: A robust and accurate toolkit/software for simulating engine dynamics,” Proceedings of the ASME 2018 Internal Combustion Fall Technical Conference, November 4–7, 2018, San Diego, CA.
- “Modeling turbulent reactive flow in internal combustion engines with an LES in a semi-implicit/explicit finite element projection method,” Proceedings of the ASME 2016 Internal Combustion Fall Technical Conference, October 9–12, 2016, Greenville, SC.
- “Three-dimensional ALE-FEM method for fluid flow in domains with moving boundaries part 1: algorithm description, *Progress in Computational Fluid Dynamics* **18** (4): 199–215 (2018).
- “An adaptive finite element method with dynamic LES for turbulent reactive flows,” *Computational Thermal Sciences* **8** (1): 57–71 (2016).
- Modeling multiphase flow: Spray breakup using volume of fluids in dynamics LES FEM method,” *Numerical Heat Transfer, Part B: Fundamentals (An International Journal of Computation and Methodology)*, **72**:4, 285–299 (2017).

Comparison Matrix

Parameter	FEARCE	CONVERGE	Fluent	Forte
Ease, speed, and quality of grid generation	Easy and quick to produce a high-quality grid	Easy and quick to produce a poor quality grid	Difficult and time-consuming to produce a grid and very difficult and time-consuming to produce a high-quality grid	Very difficult and time-consuming to produce a poor quality grid (lacks robustness throughout a cycle)
<p>Comments: Creating a grid often takes a month or more with Forte and Fluent. Convergent uses a different method, one that is quick and easy. However, the Convergent software produces a poor quality grid, with accuracy suffering in the approximation near the boundary.</p> <p>FEARCE uses an immersed boundary method to overlay complex parts on a simpler grid for the engine's cylinder and ports without any concern for these parts. FEARCE's resultant approximation remains second-order accurate at the boundary and does not suffer from having very small cells with poor integration properties. Those very small cells are a disadvantage to the competitors using cut-cell methods along with moving parts, as the cells are often irregular in shape and also require the entire solution to use smaller time increments, slowing the speed of solution. The simpler underlying grid used in FEARCE can be generated in much less time than methods used by Forte and Fluent and with higher quality than Convergent, thus saving weeks in labor costs and turnaround time.</p>				
Parameter	FEARCE	CONVERGE	Fluent	Forte
Uses predictive models for spray modeling	Yes	No	No	No
<p>Comments: FEARCE's VOF method allows for predictive liquid jet breakup into a spray of droplets. Convergent, Fluent, and Forte are engineering parameter-based models that are effective once tuned to a particular engine's injector but are not predictive. FEARCE's VOF method allows for near-exact breakup, compared to direct numerical simulation (DNS), which is often used as the exact, very resolved way to represent all relevant physical scales. See Figure 6 of a high-density liquid jet breaking up in ligaments.</p> <p>See article in the Appendix, "Turbulent Reactive Flow Modeling in Engines: A Robust and Accurate Toolkit/Software for Simulating Engine Dynamics," to see a KH-RT spray model simulation of an injector test case Spray A, of the Engine Combustion Network with tuned (non-predictive) breakup parameters and the accuracy of the KH-RT solution in for a diesel injection versus experiment, besting the competitors where the penetration is either over- or under-predicted.</p>				

Parameter	FEARCE	CONVERGE	Fluent	Forte
Uses superlinear algorithm	Yes	No	No	No

Comments: Only FEARCE achieves superlinear scaling in all cases, from engine to aerodynamics to benchmark problems. No other code can do this. Although Fluent can use 360 cores, for example, with 5,500 cells or elements per core, a 2 million cell only results in a nearly 18-fold faster turnaround. This is not nearly as successful as FEARCE, where we typically run 10 to 20 thousand elements per core, and experience a 30-times speedup for a problem at the optimal number of CPUs. After optimizing for the number of CPUs, increasing the grid resolution adds little cost in solution time, meaning wall clock time doesn't change much going from 100,000 cells and 16 cores to 1 million cells and 64 cores. See Figures 4 and for more details about this parameter.

Parameter	FEARCE	CONVERGE	Fluent	Forte
Type and level representation of conservation equations	FEM Minimum accuracy of second order, achieving third order on the largest and velocity scales (advection)	FV (Finite Volume) Achieves second-order accuracy, with first order achieved often at the largest scales or sometimes at the boundaries	FV Achieves second-order accuracy, with first order achieved often at the largest scales or sometimes at the boundaries	FV Achieves first-order accuracy on advection, as well as on some cell types

Comments: It is important in engines dominated by the largest scales to have such scales represented as accurately as possible because the largest scales are used to determine or surmise what's occurring at the smaller subgrid scales when using LES (large eddy simulation) turbulence modeling. In addition, accurately capturing the advective velocity is extremely important when determining where the reactants will be at any given time, which is critical to producing an accurate combustion simulation.

Turbulence is an important factor, but this phenomenon is smaller in scale than, say, the advection (velocity) scale. Larger-motion scales, or velocity scales, are better represented using FEARCE at a minimum of third-order accuracy without oscillatory numerical noise and essentially no diffusion or smoothing. For example, a wave front remains true to its form during propagation. The system is referred to as a stabilized FEM method. This type of accuracy and lack of diffusion on transported media is perhaps more difficult to do with Finite Volume (FV) forms or other forms of solution, and yet it doesn't usually yield the same accuracy as FEM.

The often-cited difference between FEM and FV is the use of a weak form of the governing equations, which ensures continuity of flux in FEM, whereas in FV's strong form in which

continuity is made directly. Many scientists have suggested that FEM does not locally conserve flux, an argument long laid to rest, but one that nevertheless continues to be raised even today. Nonlinear terms in FEM do require stabilization to remove numerical dispersion. Those terms in FV methods also require special handling, however, and FEM systems do this with an accuracy that beats the accuracy of general approximation.

FEM's strength lies not only in its deep foundation in mathematical theory of real analysis and vector space upon which it can draw but also in that it represents media in a continuous manner, unlike FV, which amounts to a discontinuous, piecewise representation of the material under consideration. FEM allows for good subgrid-scale interpolation and incorporation of forces at the subgrid scale to affect the grid scale and useful in point loads, such as during spray modeling. In fact, FEM spray modeling is grid convergent. That is, it has the same convergence as fluid on the grid. However, FV methods and spray modeling require fine grid resolution to obtain good results from the engineering spray models (because FV is discontinuous).

Parameter	FEARCE	CONVERGE	Fluent	Forte
Measures error in the simulated system	Yes	No	No	No

Comments: FEARCE employs an automatic grid-refinement process that increases or decreases the resolution based on error analysis. Part of the error analysis that FEARCE conducts consists of looking for the biggest changes in the shortest of distances where an error is often the largest. Both FEARCE and competitive software use a gradient method system. Such an unstructured grid enables local refinement. All competitors except for FORTE can refine just a single element. FEARCE also can use a residual error and a geometric norm to refine the grid based on the error in the solution. The FEM in FEARCE enables it to evaluate error. In other words, if the error is bounded, FEARCE determines the bound. Requiring that bound to be less than some value forces the resolution to increase in areas in which there is the largest error measure. Competitive software systems cannot measure error in the simulation system except for the unsophisticated infinity norm. Moreover, FEARCE alone can tap into other error measures, other sophisticated norms, and even patch-based error measures (such as a stress error measure). This information tells us how convergent FEARCE's solution is compared with the true solution. As such, FEARCE allows optimal resolution of the grid.

Parameter	FEARCE	CONVERGE	Fluent	Forte
Provides both AMR (h-adaptive) and Higher Order (HO) approximation (p-adaptive)	Yes	No	No	No

Comments: Only FEARCE provides both automatic grid refinement (AMR or h-adaptive) combined optionally with a Higher Order (HO) approximation (p-adaptive), where AMR refers to the finite-volume form of the grid refinement and the h-adaptive is the FEM version.

The HO method with FEM is “p-adaptive,” meaning to increase the order of the approximating orthogonal polynomials (basis or shape functions). Other software simply does not have the ability to increase the order of the approximation, making them merely second order in convergence at best. The h-adaptive refinement is driven by the sophisticated error measuring occurring while the system processes.

As discussed previously, FEARCE employs an automatic grid-refinement process, which increases or decreases resolution based on error analysis. Moreover, FEARCE can increase the order of approximation. In other words, FEARCE can increase or decrease the order of the polynomial used to approximate variables that must be solved as the solution is running.

The real benefit with FEARCE is that it can combine h- and p-adaptation. Although grid refinement produces only a second-order convergence, it does so at a low computational cost. Such accuracy is typical among the competition. By adding HO approximation, FEARCE produces exponential convergence on the order of the approximation, with HO itself pretty expensive in terms of computational cost and solution speed. However, combining h- and p-adaptive methods enables FEARCE to provide exponential convergence, with grid resolution attained with much less computational cost.

See Figure 3, which shows hp-adaptive on the same convergence curve as p-adaptive but with less computational cost and time spent during the solution process.

Parameter	FEARCE	CONVERGE	Fluent	Forte
Methods used to model turbulence	k- ω SST (Shear Stress Transport) Vreman dynamic LES	k- ϵ RNG (Re-Normalization Group) k- ϵ Smagorinski dynamic LES	k- ϵ RNG k- ϵ dynamic LES, k- ω	k- ϵ RNG k- ϵ

Comments: Only FEARCE uses Vreman dynamic Large Eddy Simulation (LES), an implicit method that automatically accounts for stress related to the damping near walls and when viscosity dominates over turbulence for energy dissipation. The Vreman enables FEARCE to transition automatically between laminar and turbulent flow, as such removing problems associated with unsteady turbulence and flow near the walls. FEARCE’s ability to use Vreman dynamic LES is ideal for simulating engine performance, where flow varies from nearly quiescent to highly turbulent 1,000s of times per minute.

Conventional LES models time-dependent or unsteady flow. Vreman LES is ideal for wall-bounded flows, unlike LES schemes that use special boundary treatment not actually based on the theory of unsteady flow. Thus, competitors would likely use a wall function near the walls to evaluate stresses there or link LES to a two-equation turbulence model that is by nature time averaged, thus removing the idea of unsteady flow violating the assumptions and physics found within an engine.

Parameter	FEARCE	CONVERGE	Fluent	Forte
Simulating the motion of immersed parts	Overset surfaces via FEM formulation for second and higher accuracy or interpolation for speed of solution	Cut-cell interpolation	Cut-cell interpolation	ALE

Comments: Arbitrary Lagrangian Eulerian (ALE) solution is generally not robust for engine modeling, even though it has been used for this purpose for more than 40 years. Grids with many moving parts that travel great distance require extra care and work during the grid-generation process when using ALE. Even then, the grid/solution combination often fails for highly resolved grids near valves, requiring the solution to be remapped or smoothed onto an adjusted grid. This mapping isn't always robust and takes extra computational time. Moreover, it requires the user to know when to call for manual grid adjustments. ALE's accuracy can also degenerate as the solution becomes less calculable or integrable on distorted grids.

Although the cut-cell interpolation methods greatly reduce the time it takes to produce a grid for the engine simulations, it does produce some odd element (cells) at the boundaries of the parts or cylinder wall. Some of these elements are very small, requiring a reduction in time-step size. Depending on the techniques used, some cells in the cut-cell method cannot always be integrated with high accuracy or the boundary is not properly represented. In addition, ability to achieve higher order accuracy is not easily possible without writing entire new discretization schemes that are rather cumbersome to implement.

The limitations described above do not occur with FEARCE. Second-order accuracy is always maintained at a minimum; there are no small cell time step size restrictions that occur; and the order of accuracy is built into the choice of the interpolation.

In addition, only FEARCE uses overset surfaces. These surfaces overlay the underlying fixed grid or Eulerian grid, making it easy for grid generators to create a grid for a fixed grid. The FEM immersed boundary provides motion for the immersed parts through the fixed grid.

Describe how your product/service improves upon competitive products or technologies.

FEARCE software is an entirely new way to examine and thus improve engine performance. With FEARCE, it is possible to simulate more accurately the laws of conservation, engine turbulence, predictive spray atomization, and different and easier grid generation. The combination of methodologies and models working together in the collective known as FEARCE has the following advantages over competitive software systems:

Provides an improved projection method for single-pressure solve and mass conservation. FEARCE includes an algorithm used to calculate momentum and mass conservation at the same time. Incorporating both pressure calculation (based on the conservation of mass) and the momentum is known as the projection scheme.

All of FEARCE's competitive software systems are typical FV methods that use Pressure-Implicit with Splitting of Operations (PISO). Unlike these, FEARCE uses a single-pressure calculation per time step scheme. As such, FEARCE is not only more accurate, but it is also more computationally efficient than the PISO schemes used by the competitive software systems. Moreover, FEARCE handles incompressible and compressible flow in one system.

Yields unprecedented accuracy. FEARCE's accuracy breaks down as follows:

- Minimum of third-order accurate for the largest scale of advection, which is known as the velocity scale.
- Minimum of second-order accurate everywhere.
- Automatically provides higher-order polynomial (basis approximation).
- Automatically provides higher-resolution grid adaptation.

FEARCE measures error in the simulation. In other words, there exists a measure of error in the simulation with the FEM formulation that FEARCE can use to drive the solution resolution and order of approximation. This means that FEARCE's simulation has a "sense" of how far from the true solution it is from what's being developed during

solution. This sense, or measure of error algorithm, automatically drives the desired grid resolution and order of approximation to the chosen error level.

Accuracy of methods is critical when it comes to properly depicting the process of engine operation. By simply changing key parameters in the simulation, it is possible to accurately model mixing, heat transfer, mass transfer, momentum of fluid, representation of fluid, and composition to the right locations at the right pressures and at the correct timing.

Figure 3, below, shows FEARCE's accuracy when it comes to simulating shocks to fluid flow.

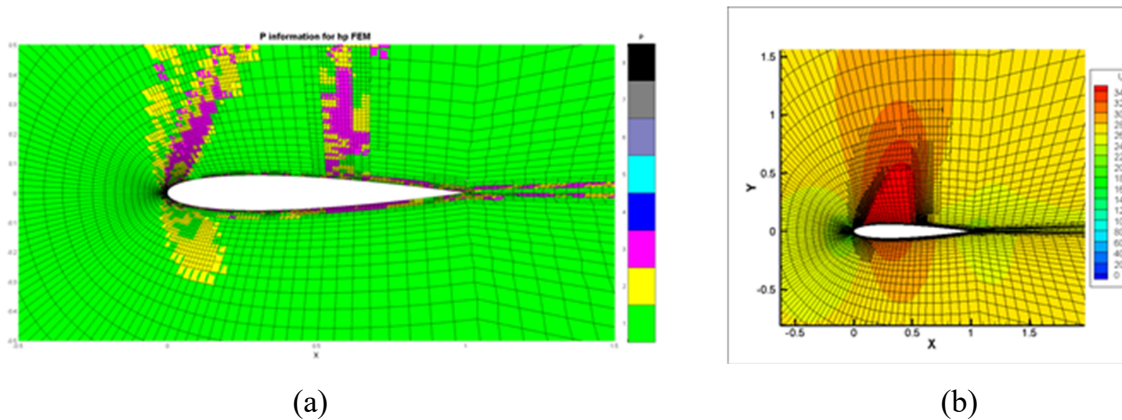


Figure 3. These plots show FEARCE's hp-adaptive system at work. Plot (a) resolves shocks created by near-sonic flow over a wing up to eighth-order approximation (the black color at the leading edge). Plot (b) shows grid resolution around the shocks' axial velocity isoachs, revealing a supersonic region between the shock regions.

Produces extremely good and superlinear parallel scaling. Parallel scalability means how well the solution speed continues to increase with larger and larger numbers of CPUs used as the problem size or resolution increases. One of FEARCE's key features is its superlinear scalability. For engine problems, FEARCE's computational time between 100,000 nodes and a million nodes barely changes when going from 16 processors to 64. In other words, just 64 processors provide for 1 million nodes because FEARCE is parallelized.

Another key advantage of FEARCE over competitive products is that it solves conservation equations, momentum, mass, and heat 3,000 times faster, thanks to its

parallel code. FEARCE even beats the superlinear curve before applying sophisticated linear equation solvers to the system of equations requiring solution.

Figure 4, below, compares FEARCE using PCG, a linear equation solver developed at Los Alamos National Laboratory. In this chart, PCG is shown without any augmentation to the matrix equations so that they are easier to solve and hence more scalable. The figure shows the scalability of the unaltered (preconditioning to make easier for iterative scheme to converge or solve) system of equations as parallelized by FEARCE.

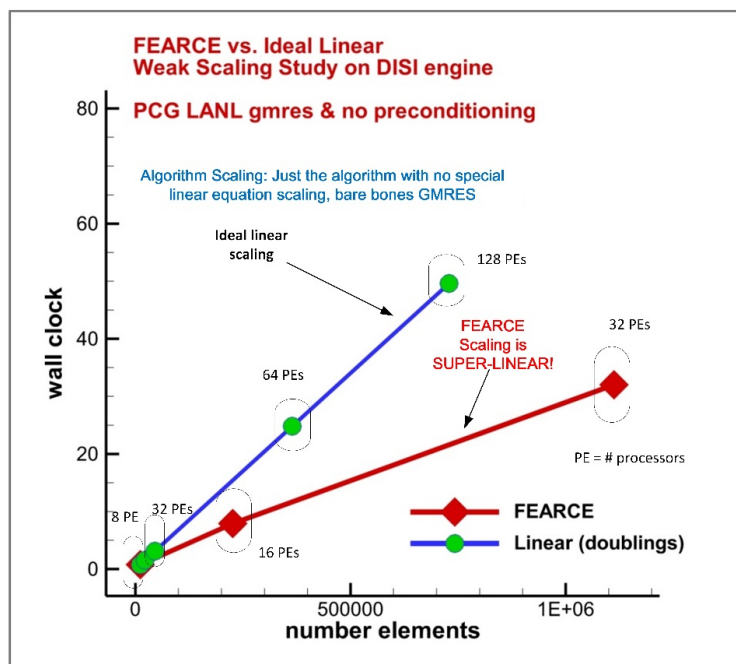


Figure 4. This chart compares linear scaling (blue line) with FEARCE's weak scaling—problem size and processors doubling simultaneously (red line)—that shows a superlinear algorithm even without matrix conditioning. Even without such conditioning, FEARCE makes the system of equations easier to solve and faster than linear scaling. (GMRES is generalized minimal residual method.)

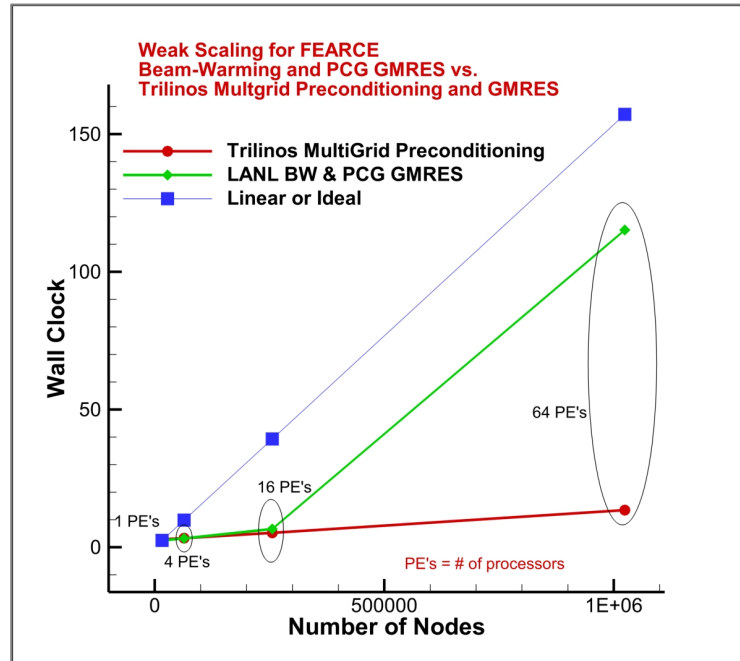


Figure 5. FEARCE uses a beam warming (BW) method for global convergence. This chart compares linear scaling with FEARCE’s weak scaling—problem size and processors doubling simultaneously (green line)—and the nearly flat scaling response (red line) when using a multigrid preconditioning technique coupled with the GMRES (generalized minimal residual method) iterative method when coupling FEARCE to Sandia National Laboratories’ open source Trilinos solver package.

Provides true multiphase flow. Unlike competitive software systems, FEARCE incorporates a VOF method that can simulate true multiphase flow, such as liquid moving through air. This feature is essential when modeling engines or other systems that contain liquid sprays.

With VOF, FEARCE can evaluate the shearing of liquid as it moves through air (and vice versa) at the liquid-air interface. This shearing force shreds liquid jets, quickly tearing the liquid stream into large droplets. This process is analogous to water exiting a garden hose. The water breaks into some larger ligaments (bigger drops), but given enough time, such bigger droplets would subsequently break down into smaller droplets. This very thing happens when fuel is injected into an engine, although there are much larger forces at work on injected fuel liquid streams that result in quickly ripping the liquid stream apart into large ligaments and then into small droplets.

Only grid resolution prevents us from using VOF to evaluate the breakup of small drops into droplets. At such a point, we switch to small particles to evaluate the transport and evaporation of these very small droplets, on the order of 0.04 cm and much less, to 1/10000 of a cm. This particle transport mode—when the drops have been broken into droplets—fully atomized is quite good at modeling their fate.

Figure 6, below, shows FEARCE's multiphase flow modeling using VOF.

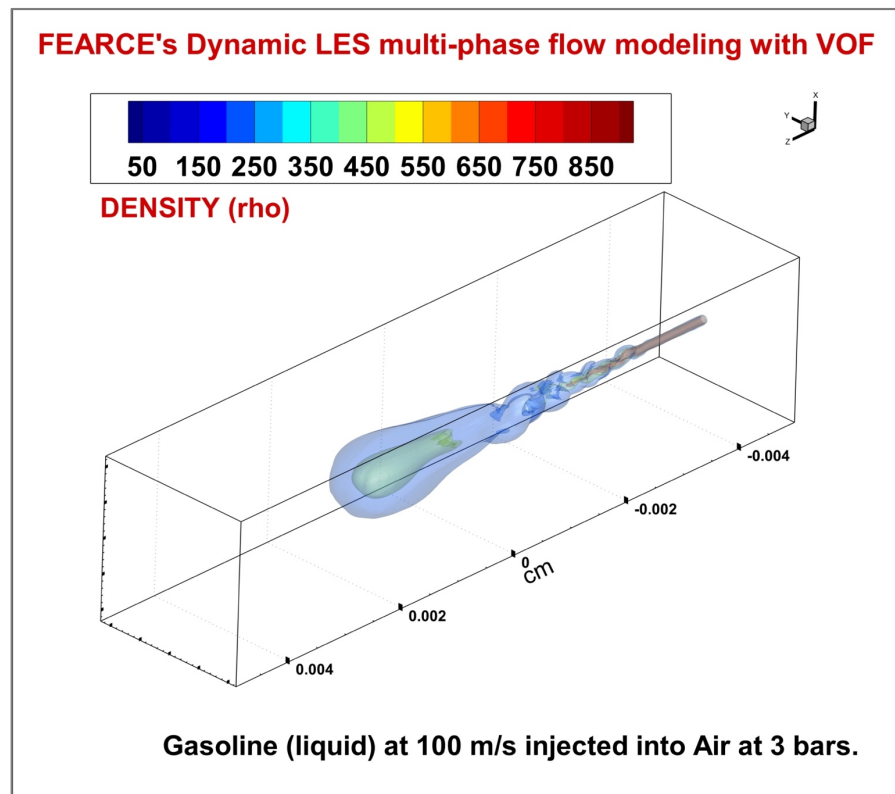


Figure 6. This figure shows liquid jet breakup in a gas-filled chamber using FEARCE's VOF three-dimensional solution. Note the very high density liquid jet breaking up into ligaments. Stresses work to tear the liquid jet apart, subsequently into droplets, if the resolution is sufficiently fine.

Generates easy- and high-quality grids. FEARCE enables the easy creation of grids (see the figure below) for an engine or other system of interest with moving parts because our software represents moving parts with a surface simply generated by computer-aided design, not part of the grid representing the engine's complex domain. Thus, the moving part's surface simply overlays the engine's fixed grid and floats through the grid by

employing an immersed boundary method to evaluate the effects of their motion on the fluid's motion and energy.

This overlaying system enables the user to generate a grid without having to consider the moving parts and fitting the grid or mesh elements around the parts, thus cutting down weeks or even months of setup time from the simulation process.

FEARCE embeds the parts after the grid generation, overlaying the gridded representation. Other software systems do not perform like this. The way FEARCE simulates moving parts allow for a grid generator to produce high-quality grids (Figure 7). (FEARCE is configured to use a grid generator, such as GridPro or CUBIT.) Not only does FEARCE produce accurate representations of moving and fixed boundaries, it does not degenerate like other software systems used to model engines.

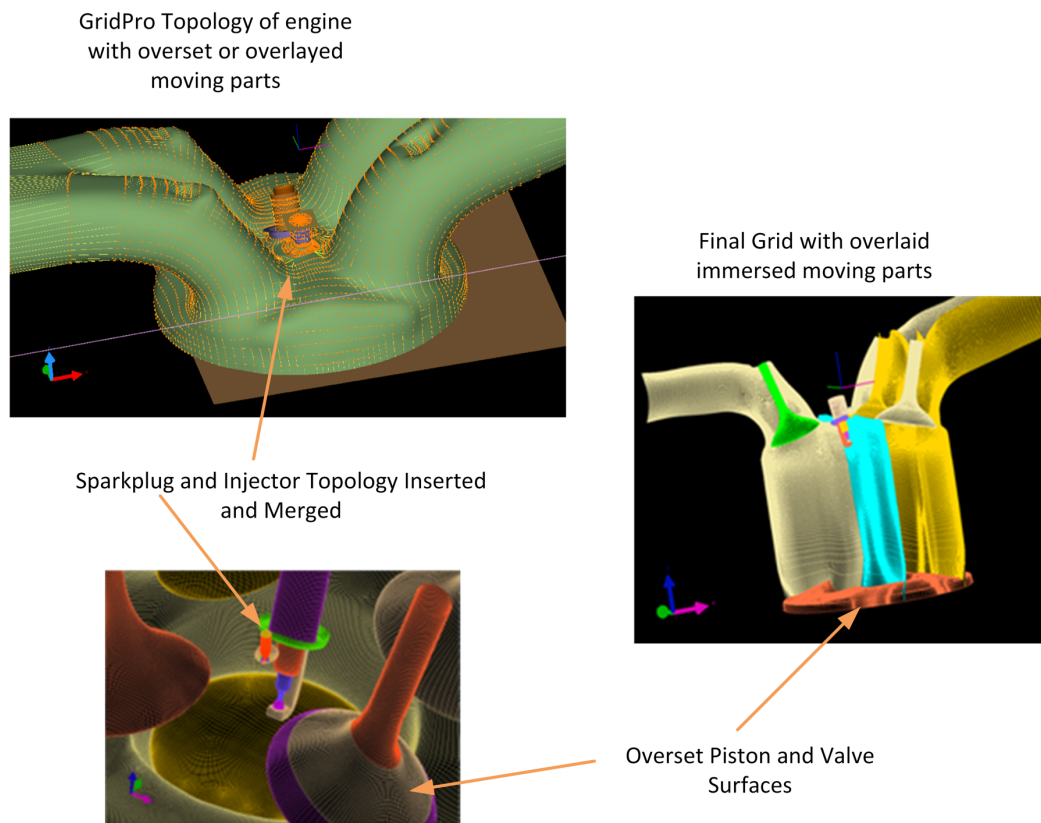


Figure 7. These images show the grid-generation process in which spark and injector-module topologies merge with the engine topology, grid, and the overset moving parts (represented by a triangular surface grid on the parts).

Describe the limitations of your product/service.

The principal limitation of FEARCE is that this software system has not been distributed widely and thus is not yet widely used. To increase the use of software and create a distribution method and foster user support, LANL issued a Request for Information [RFI] to support the broad commercialization of the software (please see the Appendix for a copy of this RFI). Commercialization plans are being finalized.

Heat transfer at an engine block or combustion chamber walls is the largest source of uncertainty when modeling the physics of combustion engines. We plan to add radiation heat transfer modeling and heat transfer through the surface to the surrounding engine with a boundary element method (BEM).

Employing the BEM method for the engine block will remove the need to grid the entire engine system with FEM, only requiring points to be placed on the CAD surface, which are then all connected to their nearest neighbors to form what is called a surface element. This will speed the turnaround time in design processing and setup time for simulations, only needing to place points on the boundary. We plan to couple the FEM scheme that models the fluid and energy dynamics within the combustion chamber in a fairly seamless manner.

Summary

Los Alamos developed FEARCE (Fast, Easy, Accurate and Robust Continuum Engineering) multiphysics software that allows engine designers to better understand turbulence inside an engine from a laptop to a supercomputer. FEARCE improves the understanding of turbulence and how it affects combustion efficiency of fuel mixing with gases to create optimum energy and efficient vehicle power.

FEARCE models engine components and runs simulations on how components influence and improve the mixing process, adjusting turbulence inside an engine to generate more power with less fuel while releasing fewer pollutants. FEARCE allows researchers to reduce expensive physical experiments, by planning better engines with its improved modeling and prediction software tools. FEARCE software can optimize and design new engine designs for use with alternative fuels.

FEARCE could help make possible to achieve gas mileages greater than 50 mpg. Such improved vehicle gas mileage could save more than 4 million barrels of oil per day, according to Robert Carling, director of the Transportation Energy Center, Sandia National Laboratories. Better-performing engines could significantly decrease vehicle greenhouse emissions and other pollutants.

Support Letters

- Michael R. Weismiller, U.S. Department of Energy Vehicle Technologies Office
- Juan C. Heinrich, University of New Mexico School of Engineering, Department of Mechanical Engineering
- Peter R. Eiseman, GridPro Program Development Company
- Peng Zhao, Oakland University Department of Mechanical Engineering
- Darrell W. Pepper, University of Nevada Las Vegas, Department of Mechanical Engineering
- Haiwen Ge, Texas Tech University, Department of Mechanical Engineering

Appendix: Supporting Information

- “Don’t forget the combustion engine,” *The Washington Post*, October 2011
- “Request for Information for FEARCE: An extensible finite element Multiphysics software suite for complex and reactive modeling.”
- “2018 FEARCE Development: A Robust and Accurate Modeling Software”
- “Turbulent reactive flow modeling in engines: A robust and accurate toolkit/software for simulating engine dynamics,” Proceedings of the ASME 2018 Internal Combustion Fall Technical Conference, November 4–7, 2018, San Diego, CA.
- “Modeling turbulent reactive flow in internal combustion engines with an LES in a semi-implicit/explicit finite element projection method,” Proceedings of the ASME 2016 Internal Combustion Fall Technical Conference, October 9–12, 2016, Greenville, SC.
- “Three-dimensional ALE-FEM method for fluid flow in domains with moving boundaries part 1: algorithm description, *Progress in Computational Fluid Dynamics* **18** (4): 199–215 (2018).
- “An adaptive finite element method with dynamic LES for turbulent reactive flows,” *Computational Thermal Sciences* **8** (1): 57–71 (2016).
- Modeling multiphase flow: Spray breakup using volume of fluids in dynamics LES FEM method,” *Numerical Heat Transfer, Part B: Fundamentals (An International Journal of Computation and Methodology)*, **72**:4, 285–299 (2017).

Principal investigator from each of the submitting organizations

PI name: David B. Carrington

Title: Scientist

Organization: Los Alamos National Laboratory

Email: dcarring@lanl.gov

Phone: 505-667-3569

Full development team member

Team member name: Jiajia Waters

Title: Scientist

Organization: Los Alamos National Laboratory

Email: jwaters@lanl.gov

Phone: 505-667-4603

Marketing and media information

Contact person to handle all arrangements on exhibits, banquet, and publicity.

First name: Janet

Last name: Mercer-Smith

Title: R&D 100 Coordinator

Organization: Los Alamos National Laboratory

Email: merc-smith_janet@lanl.gov

Phone: 505-665-9574

Contact person for media and editorial inquiries.

First name: David

Last name: Carrington

Title: Scientist

Organization: Los Alamos National Laboratory

Email: dcarring@lanl.gov

Phone: 505-667-3569

Company logo



LANL LinkedIn profile URL

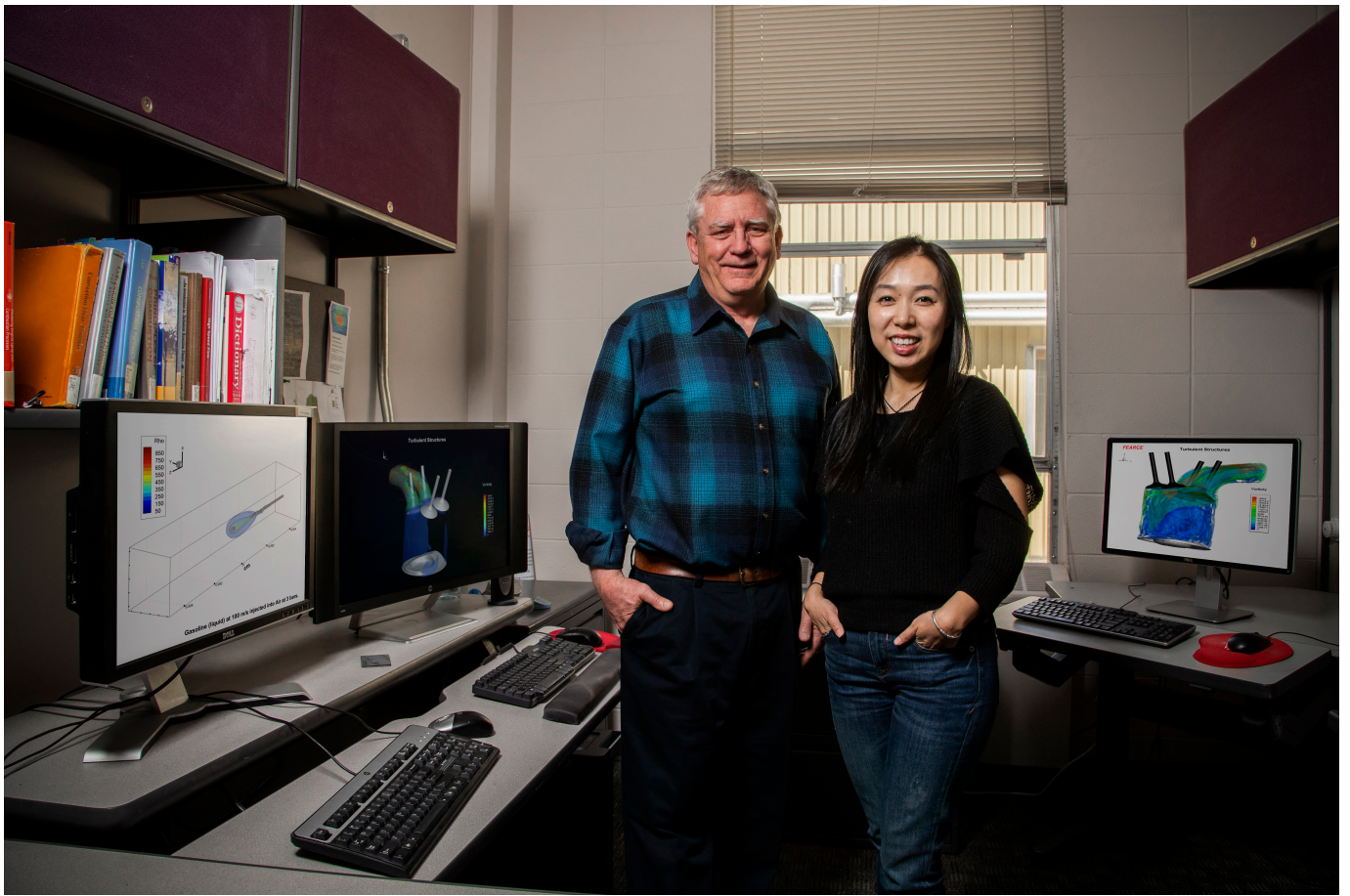
<https://www.linkedin.com/company/los-alamos-national-laboratory>

LANL Twitter handle

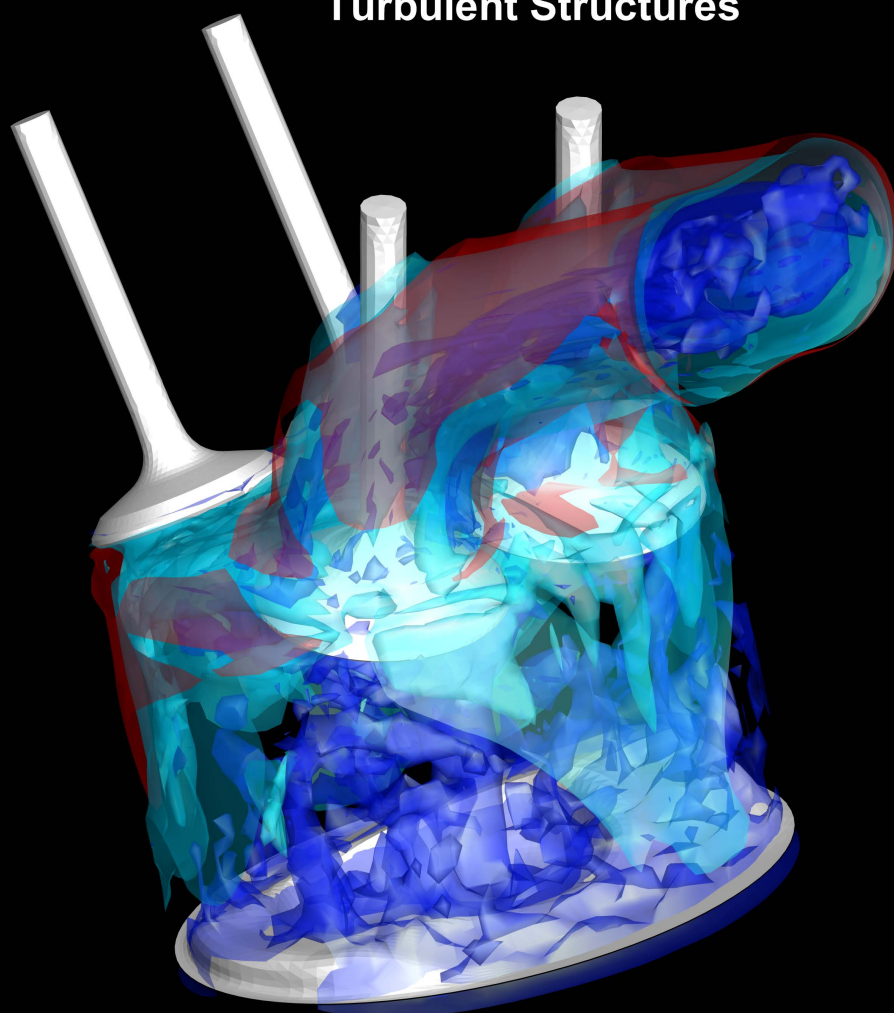
<https://twitter.com/LosAlamosNatLab>

LANL Facebook page URL

<https://www.facebook.com/LosAlamosNationalLab>



Turbulent Structures



Vorticity (1/s)





Department of Energy
Washington, DC 20585

March 11, 2019

Dear R&D 100 Committee,

I would like to extend my support for the FEARCE (Fast, Easy, Accurate and Robust Continuum Engineering) software as a recipient of the R&D 100 Award. This software was developed by Los Alamos National Laboratory (LANL) and represents a significant development in the Department of Energy's efforts to improve the understanding of internal combustion engine designs to meet the future needs of industry and researchers.

In my current role as a Technology Manager overseeing the DOE's Vehicle Technology Office's (VTO) Advance Combustion Engine portfolio, I have witnessed the FEARCE software mature into a highly valuable science and technology tool for high fidelity simulations of the internal combustion engine. FEARCE is readily and efficiently addressing the big challenge inherent with engine simulation and modeling efforts. In recent years, computational engine modeling and simulation has become an indispensable resource which plays a prominent role in our ability to design more efficient internal combustion engines and with decreased emissions.

FEARCE builds upon many years of VTO's investments in computational engine modeling, which is seen as a critical component in our ability to help improve the efficiency of internal combustion engines. FEARCE helps tease apart the enormous complexity of engine combustion processes, and allows experts to interpret and utilize the results in design processes. This is not a trivial task, as FEARCE tackles several problems facing today's engineers and scientists in a way that is more accurate and robust than other solution methods. FEARCE provides a much broader and more algorithmically complex solution to the increasingly difficult requirement being asked of vehicle engines today. The FEARCE software has been designed to perform some of the most sophisticated analyses available for these complex engine simulations. Engineers rely on software for more rapid analysis, interpretation and assessment of their designs, of which FEARCE will become a main contributing tool.

In summary, the FEARCE software is one of the truly innovative technologies that I've had the privilege to be associated with. I do not hesitate to give it my highest recommendation for an R&D 100 Award.

A handwritten signature in black ink, reading "Michael R. Weismiller".

Michael R. Weismiller, PhD
US Department of Energy
Vehicle Technologies Office
1000 Independence Ave., SW
Washington, DC 20585
(202)586-0952



Department of Mechanical Engineering

February 27, 2019

Dear R&D 100 Award Referees

I am pleased to write this letter to express my full and unconditional support to the Los Alamos National Laboratories submission of the project FEARCE to the R&D 100 award process. I am currently retired from academe after 35 years as a researcher in the area of Computational Mechanics that included 10 years at the University of New Mexico where I was Chair of the Mechanical Engineering Department.

The FEARCE software is a state of the art multi physics program designed to model turbulent reactive flows of great importance in the technical design of industrial products, most prominently internal combustion engines that demand ever increasing requirements of efficiency and lower emissions.


The role of numerical simulations in the design of internal combustion engines has increased dramatically in the past decade; the models are now used in almost every aspect of the design process. There are other software products in the market that offer capabilities for this type of simulations, e.g., Kiva, Convergent, Fluent and others. However, all of these have significant drawbacks and limitations mostly due to their reliance on older numerical technologies that cannot be readily adapted to incorporate the more efficient new methods. FEARCE incorporates the latest state of the art, some but not all of its advantages are that it is based on a Finite Elements formulation, an immersed boundary method of grid generation, adaptive mesh refinement, parallel scaling with superlinear capability, it incorporates a Volume of Fluid method to treat multiphase flow and Large Eddy Simulation of turbulent flows and a number of other numerical techniques that make the model state of the art and of improved efficiency and accuracy. The model offers true second order accuracy and very efficient equations solvers that greatly improve the speed of the simulations.

FEARCE offers a sophisticated model for the simulation of turbulent reactive flows that is relatively easy to use for the engine modeling practitioners. It offers robust and efficient mesh generating capabilities and incorporates most if not every physical phenomena such as combustion, atomization, radiation, evaporation, turbulence and other physical processes in a way that makes it possible for other users to modify and improve despite the overall complexity of the code.

When all these capabilities are put together it is easy to see that a model of this sort is an invaluable tool for the design of internal combustion engines and the great potential

payoff from its full development is hard to overstate. I want to express my full support to its entry in the 2019 R&D 100 Awards process.

Sincerely

A handwritten signature in blue ink, reading "J. Heinrich". The signature is stylized with a large, sweeping initial "J" and a long, horizontal flourish extending to the right.

Juan C. Heinrich
Professor Emeritus



**Program Development Company
300 Hamilton Avenue, Suite 409
White Plains, New York 10601**

Tel. (914) 761-1732 Fax (914) 761-1735
gridpro@gridpro.com www.gridpro.com

February 26, 2019

Dear R&D selection committee:

I am pleased to support the Los Alamos National Laboratory's entry for the R&D 100 award related to FEARCE: a CFD code that accurately addresses simulations for IC engines and more. This entry represents an exceedingly significant technology advancement that provides the most accurate and efficient way to model the physics of internal combustion engines. It is by far better at this than any competitor or would be competition anywhere. By better, I mean that a user of this methodology will get the most accurate information necessary to use for design without giving up on the efficiency of the process. Currently, the field of participants in this endeavor offer such efficiency without the necessary accuracy to enhance the engineering design itself in a really significant way. What we now have is a confluence of technology between FEARCE and GridPro that provides both process efficiency and the necessary accuracy.

A key to designing IC engines is to efficiently treat various dynamically moving geometric configurations while retaining the necessary accuracy needed to reliably steer the ICE design. The technical details vary from using high quality morphing grids that can also be overset in a good way. The grid quality from GridPro is the highest possible and is retained throughout the entire morphing (unlike those from the RBF approach). Modularity also plays a big role in the process since overset and immersed boundary techniques are strategically deployed in FEARCE. Within GridPro, modularity is naturally there up front in its topology structure where various parts and their assemblies can be set up to handle any configuration of pieces in a most automated way. In addition, there is now work progressing on automated design with GridPro that communicates very effectively with the wide range of CAD systems which, of course, connect with manufacturing systems.

What we have here is a great opportunity to connect supremely effective technologies together and perfect them as we progress. This progression will lead to the most effective way to use computers to design ICE's anywhere on this planet. I am pleased to give the highest recommendation possible.

Sincerely,

Peter R. Eiseman, President



Peng Zhao, Ph.D., Assistant Professor
Department of Mechanical Engineering

School of Engineering and Computer Science
Rochester, Michigan 48309

Re: FEARCE – R&D 100 Award Nomination

To whom it may concern,

I am currently an assistant professor of Mechanical Engineering at Oakland University, working on combustion, fuel, and advanced engine combustion strategies. I am writing to provide my unreserved support for the nomination of the engine modeling software FEARCE from Los Alamos National Laboratory for the prestigious R&D 100 Award. I have published extensively in top journals on combustion and energy, and was recently awarded the prestigious Bernard Lewis Fellowship by the International Combustion Institute for outstanding research in 2018. With my location in the intellectual center of automotive industry, I interact extensively with the automotive industry and I am well aware of the industrial needs and extensive effort to develop efficient and reliable engine simulation tools.

Due to the substantial cost savings provided relative to physical experiments, both the academic and automotive industry rely heavily on numerical simulations for the design and prediction of new engine platforms. Engine and combustion modeling involves most physical processes that can occur in heat, mass, and momentum transfer as well as chemical kinetics. This complex physics requires as accurate of models and methods we can apply and invent to achieve more predictive modeling. The Los Alamos turbulent reactive flow researchers, modelers and simulation code developers have been traditionally leading the development of engine combustion modeling and they have succeeded in providing the engine research and development community an encompassing, robust, and easy to use software for engine modeling or simulations.

Based on the success of LANL pioneers and the recent innovative work led by *Dr. David Carrington*, this more advanced FEARCE program adopts an adaptive Finite Element Method (FEM) for the solution of turbulent reactive flow on parallel and serial computers, uses a new immersed boundary method to translate moving parts within the simulated machines, and solves multi-species fluids using both RANS ($k-\omega$) and dynamic LES models for turbulent reactive flow. Various advanced submodels and features have been developed and incorporated in it, including spray and injection modeling, solution speed-up schemes, moving part treatment, spark ignition models, conjugate heat transfer, soot formation, detailed chemistry and transport, etc. FEARCE has been extensively validated on various benchmark problems to isolate various physical and chemical processes involved in engine and turbulent reacting flows. In my opinion, FEARCE represents *state-of-the-art* development in engine combustion modeling tool and is *more robust* and *more accurate* than current engine simulators, providing quick turn-around required by

design engineers. Because of these advantages and features, I anticipate that every automotive company will be adopting some form of FEARCE as part of their design-test-analysis process.

As an educator and researcher, I have seen that FEARCE is a toolbox that can be applied to many other branches of science and engineering. The comprehensive physics and chemistry contained in the code allow it to be generally useful in alternative energy infrastructure, aerospace, nuclear engineering, plasma science, astrophysics, and biofluid systems, etc. This robust and accurate systems can be used to guide the design and optimization for the offshore electricity generation platforms, wind turbines, gas turbines, combustion in scramjet engines, fluid systems and pressure vessels in commercial nuclear reactors, plasma booster for rocket engines, evolution of supernova, blood flow and internal circulation in human body, to name a few. All of the applications also have the same need for model validation and, similar to the automotive engines, FEARCE can significantly improve the computational efficiency and accuracy of such assessments for these engineering systems.

When we use FEARCE in our own research, it was immediately apparent this advanced computational platform is a unique tool that is superior to other existing tools. It is clear that FEARCE will have a significant impact on reducing computational cost and improving the accuracy and robustness of all types of reacting flow simulations, which is exactly the type of technical breakthrough that R&D 100 awards are noted for recognizing.

I hope the information above is helpful and sufficient for your consideration. Should you have any questions or require further details, please do not hesitate in contacting me.

Sincerely yours,

A handwritten signature in cursive script, reading "Peng Zhao".

Peng Zhao, PhD,
Assistant Professor of Mechanical Engineering
Oakland University
Date: Feb 6, 2019



February 24, 2019

Dear R&D 100 Award Selection Committee

I am pleased to offer my strongest endorsement of FEARCE for the R&D 100 nomination. I have been involved in both industrial as well as academic environments for over 50 years dealing with the development and application of Computational Fluid Dynamics software and related transport modeling. The success of the advanced efforts at LANL in creating FEARCE combustion software is an outstanding example of dedication and resourcefulness towards creating a state-of-the-art model, and clearly worthy of nomination for an R&D 100 award. FEARCE addresses greater accuracy in modeling combustion processes, progress toward even more efficient engines, and increased ease and speed of modeling.

Just one example of the major strengths of FEARCE is that it is entirely modular. The structure of the code allows the upper sections, e.g. physics models such as Navier-Stokes equations, turbulence models, spray models and reactive chemistry, to access the underlying fundamental data structure and operators. Researchers are not limited in the equations they need to solve. For example, one can write an equation to solve for enthalpy with access to the data structures and operators, and then send it off to the solver for solution. If one wants to add a turbulence model, the template system allows the researcher to write the model with access to the underlying data and operators, wall layer calculations and structures without having to formulate statements requiring the internal structure of operators within the code. This process is known as WYSIWYG coding - easy to see the equation directly, but not needing to know just exactly how the operators work.

The code development for FEARCE is strongly tied to the needs of its customers. Many of the key factors include:

- (1) a higher degree of accuracy than any other code of this type,
- (2) a non-cumbersome process to add submodels of nearly any nature,
- (3) access to the source code without having to duplicate the mathematical operators,
- (4) faster ability to generate grids by invention of new techniques for moving parts,
- (5) a more robust code by invention and inability for the mesh to tangle as parts move,
- (6) much faster execution times than any of its predecessors.

Since FEARCE is distributed with libraries of operators along with many types of utilities and functionals, users need only to add their specific module to a compile line, thereby allowing access to the underlying system that is then constructed directly into their executable code. This is a major difference from most commercial industry procedures.

When one considers all of these advantages, it is clear that FEARCE has significantly improved the way combustion modeling and analysis will be performed in the engine industry – leading to a major increase in engine efficiency and performance. New engine designs and enhancements will definitely stem from the application of this most unique and powerful program. The innovation and creativity of the LANL team in creating a very robust and powerful code certainly warrant making FEARCE a most worthy candidate for an R&D 100 Award.

Sincerely,

A handwritten signature in black ink that reads "Darrell W. Pepper". The signature is written in a cursive, slightly slanted style.

Darrell W. Pepper
Professor of Mechanical Engineering



TEXAS TECH UNIVERSITY

Edward E. Whitacre Jr.
College of Engineering

Department of Mechanical Engineering

Haiwen Ge, D.Sc.
Mechanical Engineering Department
Texas Tech University
2703 7th Street, Lubbock, TX 79409-1021
Phone: 608-695-5372
Email: haiwen.ge@ttu.edu

To Whom It May Concerns,

I am currently an Instructor in the Mechanical Engineering Department of the Texas Tech University and the President of Virtual Thermal Fluids, LLC. I am writing this letter to support the Los Alamos National Laboratory entry into the R&D 100 Award process entitled "FEARCE". My prior appointments at Ford Motor Company, Fiat-Chrysler-Automobiles, and John Deere focus on development of CFD-based mathematical models, simulation and optimization methods for engine combustion. I used both in-house open-source CFD code and commercial CFD software for model development and implementation. Virtual Thermal Fluids, LLC is a consulting company providing engineering service for engine development using CFD. It is my honor that I have contributed to the FEARCE by implementing spray models and chemistry solvers into it. All of this experience enables me to provide accurate evaluation of this CFD tool and make this endorsement.

Most of the CFD software available in the market are based on the finite volume method, which assumes homogeneous distribution of all quantities within each cell. It is the zeroth order in accuracy on this regard. Instead, the FEARCE features an advanced CFD solver using finite element method. Comparing to conventional finite volume method, the finite element method assumes that the quantities follow certain distribution within each cell. Usually linear shape function is employed as the FEARCE, which improves the accuracy from zeroth order in finite volume method to the first order in FEARCE. The code has been developed as a general purpose CFD tool that covers a wide range of flow regimes, including turbulent flow, reactive flow, multiphase flow, and coupling with solid simulations. Both Reynolds averaged Navier-Stokes (RANS) and large-eddy simulation (LES) have been implemented. Volume of fluid (VOF) method has been developed for simulations the interface between two phases in fluid flows. It is critically important for detailed nozzle flow and near nozzle flow simulations. In most of the CFD code, spray simulations start from the nozzle exit by initializing Lagrangian droplet parcels. The results strongly depend on initial conditions. With the VOF method, simulation of internal nozzle flow tightly couples with external spray dynamics, which offers much more accurate spray predictions. Advanced spray models such as Kelvin-Helmholtz Rayleigh-Taylor breakup model have implemented. Chemistry solver CHEMKIN has been implemented, which enables it to simulate most of reactive flows. The code has been parallelized using Message Passing Interface (MPI). Results show very good scalability, especially for large scale simulations.

As an open-source CFD code, FEARCE is a perfect platform for advanced mathematical model and numerical algorithm development as well as code implementation. Research institutes including universities and national laboratories can use it as a research tool to develop, implement, and validate their specific models and algorithms. With affordable license fee, FEARCE is a very good option for product development in industry, not only for big companies, but also for small and medium companies like Virtual Thermal Fluids LLC. It will shorten the turnaround time for engine simulations and enable fast prototype evaluations using FEARCE.

Thanks for your considerations! If you have any questions, please do not hesitate to contact me.

Best Wishes,

Haiwen Ge, D.Sc.

The Washington Post

Opinions

Don't forget the combustion engine

By Robert W. Carling

October 7, 2011

Alternative energy sources are clearly still under development. Consequently, whether those technologies are for transportation, heating our homes and buildings, or powering our computers, petroleum-based energy will be with us for a while.

This is particularly true with transportation energy. Amid the national discussion on the future of electric vehicles, biofuels, fuel cells and other advanced technologies, the fact remains that it will take decades before any new engine technology is ubiquitous in the transportation fleet. Consequently, none of the current options has begun to make a dent in U.S. oil consumption in the short term.

So now might be a good time to remember an advanced technology that is often forgotten, the combustion engine, and the promise it still holds for helping the United States meet its short-term objectives regarding oil consumption.

Studies have demonstrated that gains of [greater than 45 percent](#) to the thermal efficiency of gasoline engines are achievable in combustion engines, and fuel economy improvements of greater than 50 percent in our automobiles are [within our reach](#) when combined with other technical advances. Opportunities abound for combustion scientists and engineers to make even more improvements. The targets can be reached in the near term, not decades from now.

Let me be clear. The United States does need full and long-term

commitment to clean, advanced, alternative energy sources for transportation and other needs. Initiatives such as the Energy Department's Joint BioEnergy Institute (JBEI) in the San Francisco Bay Area are having real impact, and the department is also doing its part to support development of the next generation of biofuels, direct solar fuels, hydrogen fuel cells, batteries and electricity-producing renewables.

But with only modest investments, an existing infrastructure, and a laser-like focus by the nation's combustion engineers and scientists, we can continue to hone and refine the combustion technologies that have served us so well over the past 100 years.

By Environmental Protection Agency standards, the new Chevrolet Cruze Eco gets 42 miles to the gallon on highways, with some tests even reaching the 50 mpg mark. That's with a conventional gasoline engine, not a hybrid. With continued investments and research into new technical innovations such as the homogeneous charge compression ignition engine, the potential exists to save more than 4 million barrels of oil per day. That is roughly \$400 million per day in savings.

Because transportation represents such a sizable portion of oil use in the United States, we can achieve a 30 percent reduction in overall oil consumption if we can arrive at a 50 percent reduction in fuel use in automobiles and trucks. That is very doable from a scientific standpoint. The engine giant Cummins has already used advanced laser-based experiments to validate models that enabled an all-computational engine design, progress that saved substantial time and cost while providing a better engine and fewer tests. These advances are expanding.

Furthermore, the infrastructure for a fleet of vehicles based on

new, advanced technologies is in its infancy and will take years to fully develop. With liquid fuels, we have the infrastructure in place; a complete culture shift around the way we refuel our vehicles would not be necessary.

It won't be easy for the nation to follow this energy blueprint. Automotive companies are blunt in acknowledging that they can't solve the technical problems on their own. Their research and development budgets are shrinking, not growing.

Fortunately, the automotive industry is, in an unprecedented way, reaching out to universities and national laboratories to collaborate and build consortia. We already have the core resources, including high-tech tools such as powerful lasers, the fastest computing platforms known to man, and optical engines. All of these resources can and should be leveraged.

Those of us engaged in combustion science see it as our responsibility to bring the various sectors together to find collaborative solutions to our collective challenges, particularly those involving advanced liquid fuels and internal combustion engines. Significant improvements are well within our grasp.

We take the call for research into alternative energy solutions very seriously. But don't forget the combustion engine. It remains the most proven and the most cost-effective near-term method for reaching the nation's transportation energy goals.

Robert W. Carling is director of the Transportation Energy Center at Sandia National Laboratories in Livermore, Calif., and oversees the Energy Department's Combustion Research Facility.

 **68 Comments**



REQUEST FOR INFORMATION

Commercialization Opportunity



FEARCE: An extensible finite element multiphysics software suite for complex and reactive modeling



POTENTIAL AREAS FOR PARTNERSHIP

FEARCE is a modular multiphysics software package configured as a suite of fully-integrated tools that leverages custom templates and models alongside finite element analysis and computational fluid dynamics. FEARCE produces fast, high-fidelity solutions for any aspiring designer needing to model and predict turbulent, reactive, and otherwise complex fluid flows.

Los Alamos is seeking a commercialization partner in an applicable market that is capable of crafting and executing a go-to-market plan for the FEARCE Multiphysics suite, which is available for licensing.

INDUSTRY MARKET SECTORS

1. Computational Fluid Dynamics
2. Multiphysics Simulations
3. Computer-aided Engineering Simulations
4. Automobile & Auto-component Industry
5. Optimization and Analysis of Combustion Engines
6. Turbomachinery Design Verification



KEY HIGHLIGHTS OF OUR MULTIPHYSICS SUITE

1. STATE OF THE ART SIMULATIONS & MODELING

The FEARCE multiphysics suite delivers robust multi-species simulation and modeling capabilities for complex, turbulent, and reactive flows. The numerical methods selectively chosen for FEARCE provide a solid foundation from which fast, high fidelity, and precise solutions can be produced.

BENEFITS:

- ◎ **HIGHLY SCALABLE:** The FEARCE architecture enables real-time computational load balancing and super-linear scaling, delivering a significant improvement in solution speed compared to finite volume and finite difference formulations.
- ◎ **ACCURATE:** Methods unique to FEARCE enable continuous representation of the fluid or material yielding higher accuracy, *faster*, with fewer cells.
- ◎ **RELIABLE:** FEARCE offers correct solutions to all tested computational fluid dynamics (CFD) benchmarks.

2. ENHANCED UI/UX

The FEARCE software suite is easy-to-use and maintain. Its modular design permits mix and match customization of both the underlying simulation models and any auxiliary commercial off the shelf (COTS) tools or plug-ins.

BENEFITS:

- ◎ **EASE OF MAINTAINENCE:** Under-the-hood numerical methods need no adjustment to support new equations and models, and come with pre-built templates to further simplify simulation design.
- ◎ **ROBUST MESHING:** Novel handling of immersed moving parts enables fast, automatic and high-quality mesh generation. The mesh can never tangle, and is always stable. The mesh generated for FEARCE is automatically refined based on the error in the solution – a unique feature of the FEARCE solver.
- ◎ **COTS FRIENDLY:** The design interface of FEARCE allows easy integration with 3rd party COTS software solutions including grid generators, chemistry solvers, and computer aided design (CAD) tools.

3. ALL ENCOMPASSING AND EXTENSIBLE

FEARCE is a modeler's '*Swiss-army knife*' for reactive adaptive modeling. This suite renders solutions in any flow regime including highly unsteady, multiphase, and transitional flow. Multi-modular

design and robust multiphysics translate to an easy-to-use and versatile software regardless of the user's technical requirements.

BENEFITS:

- ⦿ **COMPLEX FLOWS:** FEARCE manages all flow regimes from incompressible laminar, to turbulent compressible, to hypersonic and transitional flows. The system handles true multiphase flow using a method that allows for predictive initial break-up of fuel injection sprays and liquid jets.
- ⦿ **COMPREHENSIVE:** FEARCE comes equipped with numerous models including evaporative spray, particle and soot activity, injection behavior, multi-material flow, multi-phase flow, and turbulence. Turbulence models have been carefully chosen to provide precise modeling in wall-bounded domains.
- ⦿ **CUSTOMIZABLE:** FEARCE is extensible to solid stress modeling, fluid structure interactions, porous media modeling, magneto-hydrodynamics, *and more!*



LOS ALAMOS NATIONAL SECURITY (LANS) **INTELLECTUAL PROPERTY**

International copyright(s) on C18059 - FEARCE Suite and related software.

Please note that the U.S. Government retains a worldwide, royalty-free, non-exclusive right to practice any LANS-owned patents and/or copyrighted software. Accordingly, any and all partners will have open access to any LANS intellectual property in performance of a Government contract.



PREFERRED PARTNER ATTRIBUTES

- ⦿ Demonstrated knowledge of product marketing, sales, and worldwide software distribution
- ⦿ Technology commercialization strategy and business plan (e.g., in-house development, partnering with industry leaders, sublicensing, etc.)
- ⦿ Financial and human resources to be dedicated to this commercialization effort
- ⦿ Established experience with software marketing and channel development
- ⦿ Ability to maintain software licensing and security mechanisms (or equivalent) for versatile licensing options and for appropriate protection of source and executable versions of the code(s)
- ⦿ Ability to provide/acquire support for maintenance and further development
- ⦿ Expertise in one or more of the following: computational fluid dynamics, solution of linear equation systems, parallel solution processes; and grid generation
- ⦿ Proven technical and customer support model
- ⦿ One or more U.S. executives with whom LANS personnel may interact
- ⦿ Ability and willingness to ensure compliance with U.S. Export Control law is a requirement



WHAT WE ARE REQUESTING

Please submit a [written response](#) on how your organization envisions utilizing and deploying this technology in partnership with Los Alamos. We look forward to reviewing your ideas on how we can together bring the FEARCE suite to the private sector. [Please respond by email to \[kbadura@lanl.gov\]\(mailto:kbadura@lanl.gov\)](#), or [call Kaelyn Badura directly at \(505\) 665-8032 by Friday, June 29, 2018.](#)

Los Alamos National Security, LLC (LANS) is the manager and operator of the Los Alamos National Laboratory for the U.S. Department of Energy National Nuclear Security Administration under contract DE-AC52-06NA25396. LANS is a mission-centric Federally Funded Research and Development Center focused on solving the most critical national security challenges through science and engineering for both government and private customer.

I.15 2018 FEARCE Development: A Robust and Accurate Engine Modeling Software

David B. Carrington, Principal Investigator

Los Alamos National Laboratory (LANL)
 P.O. Box 1663
 MS-B216
 Theoretical Division, T-3 Fluid Dynamics and Solid Mechanics
 Los Alamos, NM 87544
 E-mail: dcarring@lanl.gov

Michael Weismiller, DOE Technology Development Manager

U.S. Department of Energy
 E-mail: Michael.Weismiller@ee.doe.gov

Start Date: October 1, 2017	End Date: September 30, 2018	
Project Funding (FY18): \$700,000	DOE share: \$700,000	Non-DOE share: \$0

Project Introduction

Research and development of **Fast, Easy, Accurate and Robust Continuum Engineering (FEARCE)**, formerly KIVA-hpFE) for turbulent reactive and multiphase flow, particularly as related to engine modeling, is relevant to the DOE Vehicle Technologies Office efforts at addressing national energy security. Less dependence on petroleum products leads to greater energy security. By U.S. Environmental Protection Agency standards, some vehicles are now reaching the 42–50 mpg mark. These are conventional gasoline engines. With continued investment and research into new technical innovations, the potential exists to save more than 4 million barrels of oil per day, or approximately \$200 to \$400 million per day. This would be a significant decrease in emissions and use of petroleum and a very large stimulus to the U.S. economy.

Better understanding of fuel injection and fuel–air mixing, thermodynamic combustion losses, and combustion/emission formation processes enhances our ability to minimize fuel use and unwanted emissions. Helping to accomplish this understanding, the **FEARCE** or KIVA development project is providing a state-of-the-art capability for accurately simulating combustion processes: to have a predictive methodology in software helping industry and researchers not only meet national goals on fuel usage and emissions, but global goals. In addition, a predictive, robust, and accurate capability for simulating the engine combustion process helps to minimize time and labor for development of new engine technology.

Objectives

A main goal of the **FEARCE** or KIVA development project is to help provide better understanding of engine combustion processes in order to enhance the ability to minimize fuel use and unwanted emissions. The **FEARCE** development project is providing a state-of-the-art capability for accurately simulating combustion processes and is providing a more predictive methodology than currently available in software to supply industry and researchers a tool to help meet national goals on emissions and engine efficiencies. In addition, a predictive, robust, and accurate capability for simulating engine combustion processes helps to minimize time and labor for development of new engine technology. To meet these goals, our project objectives are listed as follows.

Overall Objectives

- Develop mathematical and computer algorithms and software for the advancement of speed, accuracy, robustness, and range of applicability of FEARCE, an internal engine combustion modeling software package, to be a more predictive computer code. This is to be accomplished by employing higher-order, spatially accurate methods for reactive turbulent flow and more predictive spray injection, combined with a robust and accurate actuated parts simulation along with more appropriate turbulence modeling. In addition, we seek to understand the effect of heat transfer and the variation of temperatures on the internal combustion engine by creating easy-to-use numerical methods that eliminate all usual assumptions about such phenomena, such as assumed heat transfer processes at chamber and part boundaries. The code combines state-of-the-art chemical reaction simulators, such as Chemkin-Pro.
- Provide engine modeling software that is easier to maintain and easier to add models to than the current KIVA codes, and reduce code development costs into the future via more modern code architecture. In addition, FEARCE is being developed to be a commercially available software package, where DOE and LANL are doing the very difficult longer-term research for better modeling software which is best done using the types of capabilities available at the national laboratories.
- Provide software capable of producing fast turn-around times needed by industry. The code not only functions well on small computer platforms but addresses high-performance computing aspects required for high-fidelity and more predictive solutions. These objectives require extensive use of high-performance computing, thereby requiring our work to employ modern frameworks and methods that take advantage of computer resources very effectively, which FEARCE has accomplished by scaling to the size of the problem in a super-linear manner, the holy grail of high-performance computing.

Fiscal Year 2018 Objectives

- Develop a four-valve direct injection, spark ignition (DISI) engine system for validation of FEARCE
- Validate progress of FEARCE on experimental data of the four-valve DISI engine; collaborate with Dr. Magnus Solberg of Sandia National Laboratories (SNL) on the DISI setup and experimental data
- Construct systems to use ChemKin II and III and ChemKin-Pro reactive chemistry software
- Continue spray model development for both predictive spray break-up and subsequent droplet transport and fate; implement the Kelvin Helmholtz – Rayleigh Taylor (KH-RT) spray model and perform validation against data stored on the Engine Combustion Network (ECN) website from various experimentalists
- Develop faster linear solver system by implementing a multigrid solution system of linear equations that improves our current implicit solutions methods by more than a factor of two
 - Invented a method for implementing Message Passing Interface (MPI) for today's and future platforms [1] that is super-linear
- Begin the process of commercialization of FEARCE

Approach

Our approach is founded in designing, inventing, and developing new modeling methods and software. The design is a finite element method (FEM). Many beneficial and salient attributes of the software stem from the FEM formulation. We invented and developed the following systems to date (details are provided in the referenced publications).

- Developed the FEM predictor–corrector scheme projection method for high accuracy and all the benefits the FEM system brings to computational fluid dynamics (CFD) modeling of engines [2,3]

- Developed the hp-adaptive system for higher-order accuracy, where ‘h-adaptive’ is automatic grid refinement and ‘p-adaptive’ is higher-order approximation as driven by the error measure of the simulation [4]
- Invented the local-arbitrary Lagrangian–Eulerian (ALE) method for moving bodies [5]
 - Invented a moving marker system to track any chosen interfaces and reconstruct intersected elements to match the interface
- Developed immersed boundary methods for moving bodies [6]
- Developed new dynamic large eddy simulation (LES), specifically designed for wall-bounded flows [7]
 - Self-damping turbulence at the walls negates the need for a law-of-the-wall system
- Invented and developed volume-of-fluid (VOF) methods in FEM for true multi-phase compressible flow to fully represent the spray break-up process and to have predictive spray modeling [8]
- Developed a fast linear solver system
 - Developed parallel solution method [1]
 - Delivered 30× speed-up over serial code given the same problem and settings
 - Implicit solutions methods for 10× speed-up over serial parallel for an overall 300× speed-up
 - Added Trilinos [9] Multigrid matrix solution, further improving solution speed and parallel scaling by order of magnitude (8x) over Implicit Beam-Warming system in FEARCE (that delivers 300x) for a total of 2,400x speed-up over explicit serial version
 - Delivers super-linear scalability
- Invented a method for implementing MPI for today’s and future platforms [1]

We are building models and code so that they meet all the objectives in easy-to-maintain software that easily handles addition of others’ submodels. Careful verification and validation of the methods and code is required. The development of this technology utilizes many areas of expertise, including multi-species turbulent reactive flow modeling with liquid sprays, modeling of immersed moving bodies, and the extensive numerical methods for the solution of the model and governing equations developed in the software.

Results

Our efforts this year continue to push toward a comprehensive tool for the future with the accomplishment of more grid generation improvements, validation of immersed moving parts including four-valve DISI engine, the KH-RT spray model, and an algebraic multigrid linear equation solver implementation for even greater computational speed. We’ve also begun the process of commercializing the software to be able to fully support the requirements of industries and researchers for a simulation software.

Grid Generation

- In conjunction with Program Development Company, who developed GridPro, we are working on providing high quality grids for the engine system with an eye toward ease of use. The overset parts system used in the moving parts algorithm allows for easy grid generation of the cylinders and ports, with the spark and injector modules easily inserted. The piston and valves surfaces simply are also inserted by overlaying their surface representations after a quality grid is automatically generated.
- The overset gridding greatly simplifies the gridding process, removing the need to work around immersed bodies employed in traditional gridding methods. The injector and spark systems are built separately with the idea of making various types of injectors and spark plug modules that are simply connected to the engine cylinder grid. It cannot be overstated: a quality grid is needed to produce

reliable simulations. Gridding is a major component of CFD, where we seek to provide that quality with a minimum of labor.

Engine Simulation and Continued Validation of Immersed Moving Parts for the Engine System

- We developed an immersed boundary method and are developing the immersed FEMs for moving bodies using FEARCE's surface marker system. This work is partially based on methods used in our local-ALE system for moving bodies.
- The moving marker system utilizes track-moving boundary interfaces [5].
- Immersed boundary employs interpolation and projection of nearest nodal values normal to the surface. Immersed FEM utilizes the shape or basis functions for interpolation and a projection system to place a point along the normal to the surface, from which the nearest node is projected, to calculate the fluid's motion and thermodynamic state.
- A four-valve engine test case is functioning as shown in Figure I.15.1 using the immersed boundary methods, showing turbulent flow structures (by vorticity).

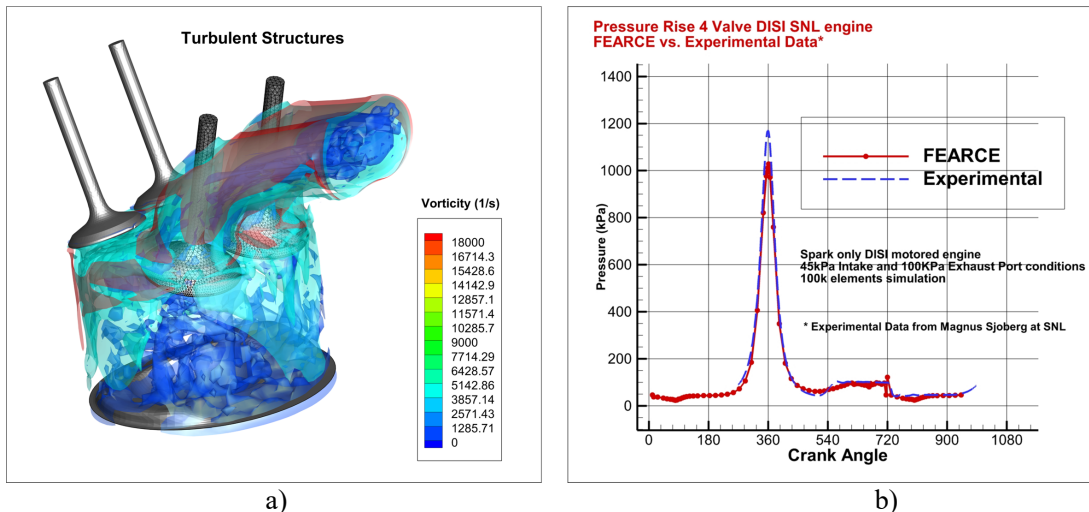


Figure I.15.1. Four-valve DISI engine: a) turbulent structures shown by magnitude of vorticity (1/s) during intake; b) pressure rise as a function of crank angle as compared to experimental data

Spray Modeling

- We implemented the KH-RT spray model into FEARCE. Tests have been conducted on Spray A and Spray G ECN test cases to date with the following results. The KH-RT spray model [10] for the Spray A case simulates injection of diesel into quiescent nitrogen at 2.2 MPa, as shown in Figure I.15.2. Figure I.15.2a shows the droplets at 2 μ s. Figure I.15.2b shows the penetration (mass moment distribution) of the spray droplets over time compared with experimental data from ECN.

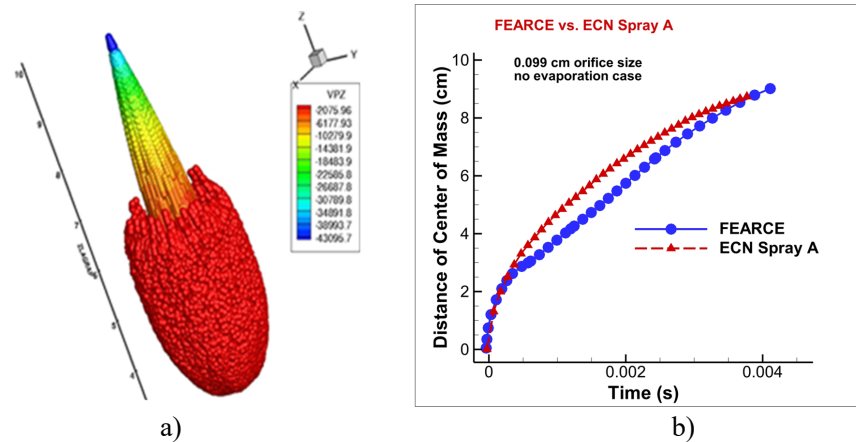


Figure I.15.2. The ECN Spray A case: (a) injection of diesel in quiescent nitrogen at 2.2 MPa, KH-RT spray model; b) the penetration depth of the spray compared to ECN experimental data

- We are developing a system for fully representing the injection process from our current predictive spray break-up process using VOF, as shown in Figure I.15.3. The system hands off the predicted spray break-up into ligaments and subsequent droplet transport modeling and evaporation, allowing true spray break-up transition to the Lagrangian particle and Rayleigh–Taylor secondary break-up systems, thereby producing more accurate engineering modeling for the injection system. Figure I.15.3 shows liquid being injected into air at 3 bar through an orifice of 0.01-mm diameter early in time. The break-up length where the wave instabilities are large enough to cause ligamentation is five orifice diameters downstream of inlet, which is near the results obtained by direct numerical simulation (DNS) as reported in Waters et al. [8].

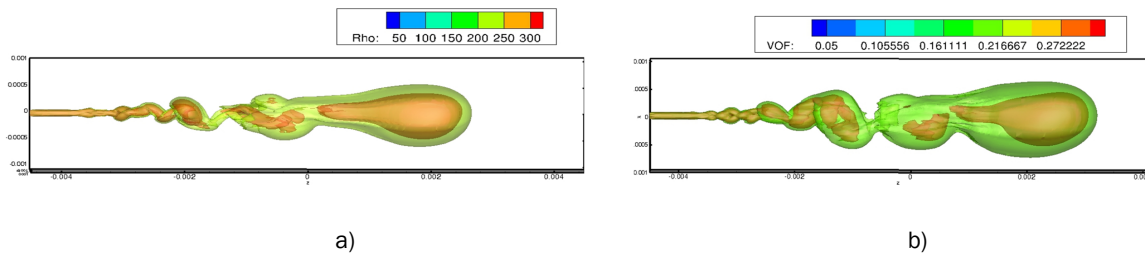


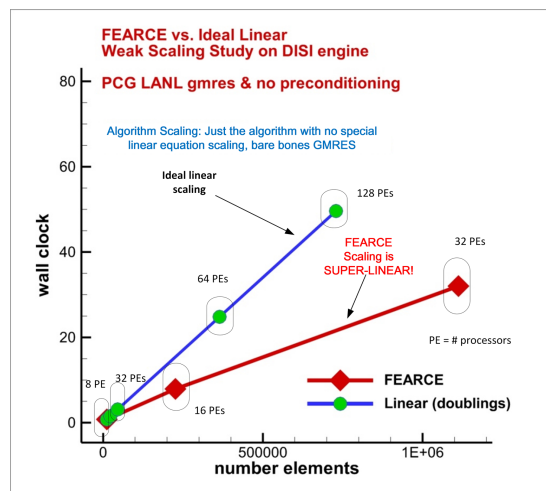
Figure I.15.3. Multiphase flow simulation with VOF method, gasoline injected into quiescent air at 3 bar: (a) gasoline jet primary break-up into ligaments and (b) primary break-up and w-component of velocity of air showing recirculation

Computational Efficiency

We continued work on parallel solution method and reducing wall-clock time by adding the Trilinos Multigrid preconditioning. Previously, we had developed a 10 \times speed-up with the implicit solve related to increased time step size. Additionally, we produced a 30 \times speed-up over the serial version with the implementation of a shared-node FEM system that reduces communication cost and produces a super-linear scaling for an over 300 \times speed-up [1]. We installed systems to access the Trilinos solver package where the

multigrid preconditioning is providing about $8\times$ speed-up for a total of $2,400\times$ speed-up over the serial version of FEARCE. Multigrid improves the already good parallel scaling when running on a large number of processors. Keeping in mind that the parallel version of FEARCE is significantly faster than KIVA-4mpi, the parallel version of KIVA-4, significant strides have been made at the speed of solution.

- We are delivering super-linear scalability, as was demonstrated in Waters and Carrington [1], in a strong scaling experiment on standard CFD benchmark problems such as the backward-facing step or flow over a cylinder. Shown in Figure I.15.4 is the scaling of FEARCE's algorithm (without special linear equation solver treatments such as preconditioning or multigrid), besting the ideal linear scaling.
- We implemented access to the Algebraic Multigrid Preconditioning and linear equation solvers from Trilinos (<https://trilinos.org/>).
 - We improved wall-clock times by a factor of 8 over our original $300\times$ speed-up, beyond $2,400\times$ speed-up over our explicit serial solver, as shown in Figure I.15.5. We are now encroaching on exceptional high-performance computing performance. Note that optimal performance usually requires some domain distribution alteration, not simply the doubling shown in the scaling analysis in Figures I.15.4 and I.15.5.
 - Further gains in the wall-clock times are expected for the super-linear system by employing greater vectorization and use of graphic processing units (use of Kukkos with Trilinos).
- It is significant to note that FEARCE requires far fewer elements to achieve the same accuracy as older KIVA codes, allowing for much faster solution on the same resolution with higher accuracy. This is the idea of high-performance computing, getting the most solution accuracy and speed from the least amount of computational work, utilizing the least of a computer and getting better accuracy, allowing for high-resolution systems having extremely good accuracy.
- FEARCE produces better accuracy than previous codes and on coarser grids. Hence, the new code is capable of being faster on the same resolution as old codes but is more accurate even on less resolved problems, providing additional advantages. Previous reports show ever increasing computational speed versus KIVA-4mpi.



GMRES – generalized minimal resolution method

Figure I.15.4. FEARCE's super-linear algorithm scaling versus the ideal scaling curve

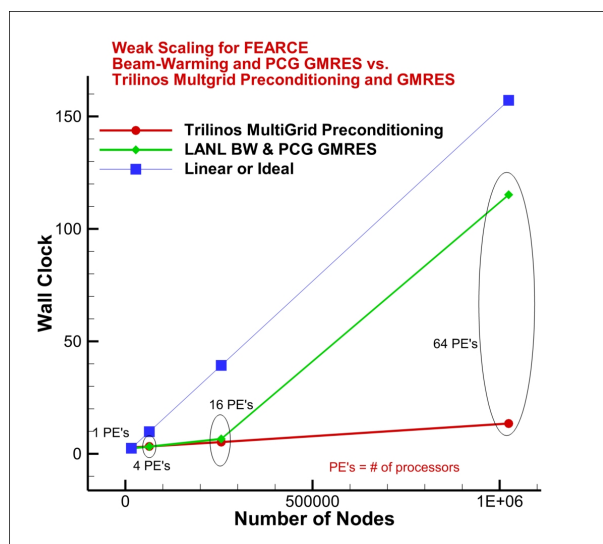


Figure I.15.5. FEARCE's Beam-Warming (BW) system versus use of Trilinos Multigrid preconditioned GMRES, a weak scaling study

Conclusions

The KIVA development project at LANL is nearing the objective of having robust state-of-the-art CFD software for turbulent reactive flow, particularly well-suited for combustion modeling in engines or machines where immersed moving boundaries are involved, all with an eye toward solutions produced on quality grids created with a minimal amount of labor.

- Fast grid generation: computer-aided drawing to CFD grid in nearly a single step
- Four-valve DISI engine experimental data used to validate the robust moving immersed FEM method
- KH-RT spray model added to the code with validation ongoing via the ECN test cases
 - Spray A case with KH-RT for validation
 - Spray G cases with evaporation proceeding
- Predictive spray modeling with the addition of VOF method
 - Developing transition to Lagrangian particle transport from predictive spray break-up for engineering-type simulations
- Highly scalable parallel solution system, with multigrid preconditioning producing nearly perfect scaling, 2,400 times faster than serial version of FEARCE, 8 times faster than just the super-linear FEARCE and only gmres Krylov linear equation solver
 - Researching Exascale possibilities by using vectorizable Cuda friendly sections of code for graphic processing nested into the MPI parallel framework
- ChemKin II/III and also Chemkin-Pro added for faster, larger, and more robust reactive chemistry

Publications

1. Carrington, D.B., and J. Waters. 2018. "Turbulent Reactive Flow Modeling in Engines: A Robust and Accurate Toolkit/Software for Simulating Engine Dynamics." *Proceedings of the ASME 2018 Internal Combustion Fall Technical Conference*, ICEF2018, San Diego, CA, USA, November 4–7, 2018.
2. Hatamipour, V.D., D.B. Carrington, and J.C. Heinrich. 2018. "Accuracy and Convergence of Arbitrary Lagrangian-Eulerian Finite Element Simulations based on a Fixed Mesh." *Progress in Computational Fluid Dynamics*, an Int. Jour., vol. 18, no. 4, pp. 215–231.

3. Carrington, D.B., M. Mazumder, and J.C. Heinrich. 2018. “Three-Dimensional Local ALE-FEM Method for Fluid Flow in Domains Containing Moving Boundaries/Objects Interfaces.” *Progress in Computational Fluid Dynamics*, an Int. Jour., vol. 18, no. 4, pp. 199–215.
4. Waters, J., D.B. Carrington, X. Wang, and D.W. Pepper. 2018. “A Dynamic LES Model for Turbulent Reactive Flow with Parallel Adaptive Finite Elements.” In *Energy for Propulsion, Chapter 3: Turbulent Combustion Modeling and Simulations*, 217–235. Singapore: Springer.
5. Waters, J., and D.B. Carrington. 2018. “Modeling Multi-Phase Flow: Spray Break-up Using Volume of Fluids in a Dynamic LES FEM Method.” 2018 AIAA Aerospace Sciences Meeting, AIAA Science and Technology Forum and Exposition 2018, Kissimmee, FL, January 8–12, 2018.
6. Waters, J., D.B. Carrington, and M.M. Francois. 2017. “Modeling Multi-phase Flow: Spray Break-up Using Volume of Fluids in a Dynamic LES FEM Method.” *Numerical Heat Transfer, Part B*, vol. 72, no. 4, pp. 285–299.

References

1. Waters, J., and D.B. Carrington. “A parallel Large Eddy Simulation in a finite element projection method for all flow regimes.” *Numerical Heat Transfer, Part A*, vol 70, n0. 2, pp. 117–131, 2016.
2. Carrington, D.B. 2009. “A Characteristic-Based Split hp-Adaptive Finite Element Method for Combustion Modeling in KIVA-hpFE.” LANL Scientific Report no. LA-UR-09-06527.
3. Carrington, D.B., X. Wang, and D.W. Pepper. 2014. “A Predictor-Corrector Split Projection Method for Turbulent Reactive Flow.” *Journal of Computational Thermal Sciences* 5, no. 4: 333–352.
4. Carrington D.B., X. Wang, and D.W. Pepper. 2014. “An hp-Adaptive Predictor-Corrector Split Projection Method for Turbulent Compressible Flow.” *Proceedings of the 15th International Heat Transfer Conference, IHTC-15*, Kyoto, Japan, August 10–15, 2014.
5. Carrington, D.B., M. Mazumder, and J.C. Heinrich. 2018. “Three-Dimensional Local ALE-FEM Method for Fluid Flow in Domains Containing Moving Boundaries/Objects Interfaces.” *Progress in Computational Fluid Dynamics* 18, no. 4: 199–215.
6. Carrington, D.B., and J. Waters. 2018. “Turbulent Reactive Flow Modeling in Engines: A Robust and Accurate Toolkit/Software for Simulating Engine Dynamics.” *Proceedings of the ASME 2018 Internal Combustion Fall Technical Conference, ICEF2018*, San Diego, CA, USA, November 4–7, 2018.
7. Waters J., D.B. Carrington, and D.W. Pepper. 2016. “An Adaptive Finite Element Method with Dynamic LES for Incompressible and Compressible Flows.” *Journal of Computational Thermal Sciences* 8 (1): 57–71.
8. Waters, J., D.B. Carrington, and M.M. Francois. 2017. “Modeling Multi-phase Flow: Spray Break-up Using Volume of Fluids in a Dynamic LES FEM Method.” *Numerical Heat Transfer, Part B* 72, no. 4: 285–299.
9. Trilinos: <https://trilinos.org/>
10. Reitz, R.D. 1987. “Modeling Atomization Processes in High-Pressure Vaporizing Sprays.” *Atomization and Sprays Technology* 3: 309–337.

Acknowledgements

- Jiajia Waters, LANL: Co-Principal Investigator
- Brad Philpbar, LANL: GRA
- Michael Weismiller and Gurpreet Singh, U.S. Department of Energy’s Vehicle Technologies Office: Managers of the Advanced Combustion Engines and Fuels research portfolio

Acronyms, Abbreviations, Symbols, and Units

KH-RT	Kelvin Helmholtz – Raleigh Taylor
FEM	Finite Element Method

VOF	Volume of Fluids
ECN	Engine Combustion Network
LES	Large Eddy Simulation
PCS	Predictor-Corrector Split

ICEF2018-9552

TURBULENT REACTIVE FLOW MODELING IN ENGINES: A ROBUST AND ACCURATE TOOLKIT/SOFTWARE FOR SIMULATING ENGINE DYNAMICS

David B. Carrington

Los Alamos National Laboratory
Los Alamos, NM, USA

Jiajia Waters

Los Alamos National Laboratory
Los Alamos, NM, USA

ABSTRACT

The Los Alamos turbulent reactive flow researchers, our modelers and simulation code developers have succeeded in providing the engine research and development community an encompassing, robust, accurate and easy to use software for engine modeling or simulations. This software is now known as the FEARCE Toolkit.

In this paper we discuss the physics present in the engine by discussion the methods we've employed to solve the model equations within the toolkit. Provided are background on what has been developed recently at LANL for internal combustion engine modeling.

INTRODUCTION

Engine combustion involves turbulent flows and a variety of complicating factors. These factors include highly nonlinear chemical kinetics, small-scale velocity and scalar-mixing, turbulence–chemistry interactions, compressibility effects (volumetric changes induced by changes in pressure), and variable inertia effects (volumetric changes induced by variable composition or heat addition). Coupling between these processes occurs over a wide range of time and length scales. Further complications arise when multiple phases are present due to the introduction of dynamically evolving interface boundaries and the complex exchange processes that occur as a consequence. In the calculation of turbulent flows, the time-averaged Navier-Stokes equations are widely used as governing equations, and this may give good results for mean velocity and pressure fields by using various turbulence models (*i.e.* $k - \omega$) for turbulent motions. However, it is evident that the RANS approaches are fundamentally unable to capture detailed flow behaviors and particularly the unsteady turbulent structures. Therefore, combustion is notoriously difficult to model at the Reynolds-averaged level. LES is rapidly becoming more widely used to study combustion in many modern combustion devices [1-6]. But turbulence modeling is not the end all in developing an encompassing tool; the largest scale in momentum and species transport for engines is simply the advection of the mass.

Simulating this accurately is a must, and this is discussed too in the following methods we employ in the new code.

The new FEARCE Toolkit system is an adaptive Finite Element Method (FEM) for the solution of turbulent reactive flow on parallel and serial computers and was developed for engine and combustion modeling [6-10]. The newly developed adaptive Finite Element Method (FEM) flow solver uses the 'h' adaptation providing for grid refinement and the 'p' adaptive system providing higher order approximation [11-13]. The system uses our new immersed boundary method (IBM) to translate moving parts within the simulated machines, part of which is based on the original local-ALE method [14-16]. This new FEM solver is parallel (MPI), solves multi-species fluids using both RANS ($k - \omega$) and dynamic LES models for turbulent reactive flow [6-10]. The code consists of many modules, including the following main portions known as the Engine Modeling Systems (EMS) modules:

EMS	(Reactive Adaptive Modeling, core FEM method)
EMS – AFWNS	(All Flows Navier-Stokes)
EMS – HOFE	(Higher Order Finite Element)
EMS – Imbound	(Immersed Moving Boundary)
EMS – MPMM	(Multi-Phase, Multi-Material)
EMS – InjectaS	(Injection, Spray and Soot)
EMS – RANS	(Reynolds Averaging Navier-Stokes)
EMS – LES	(Large Eddy Simulation)

NOMENCLATURE

\sim	designates a Favre-averaged variable
$-$	designates a grid-filtered variable
c	sound speed (m/s)
C_p	specific heat capacity at constant P (J/kg.K)
C_{vm}	Vreman fixed SGS eddy viscosity coefficient
C_{DVMG}	Vreman dynamic SGS eddy viscosity coefficient
D_j	diffusion coefficient of the j^{th} species (m^2/s)
D_k	turbulent diffusion coefficient (m^2/s)
E	total internal energy (J/kg)
f_k	body forces (N/m^3)

f_{drop}	body forces related to particulate or droplets in flow (N/m^3)
H_j	enthalpy of species j (J)
H_{oj}	enthalpy of formation (J)
P	pressure (Pa)
Pr	molecular Prandtl number
Pr_{sgs}	SGS eddy Prandtl number
Pr_{DVMG}	Vreman dynamic SGS eddy Prandtl number
Q_j	subtest-scale heat flux vector
q_i	heat flux vector
Re	Reynolds number
\tilde{S}_{ij}	strain rate tensor
Sc	Schmidt number
Sct	Subgrid scale turbulent Schmidt number
T	temperature (K)
T_{ij}	subgrid test-scale stress tensor
t_{ij}	grid scale shear stress
u_i	velocity component (m/s)
Y_{jfj}	body force term for the j^{th} component
\dot{w}_{chem}^j	chemical reaction
\dot{w}_{spray}^j	spray evaporation

Greek Symbols

∂t	discrete time step size (s)
κ	Coefficient of thermal conductivity (W/m·K).
ρ	density (kg/m^3)
Y^j	mass fraction (jth species)
τ_{ij}	subgrid scale stress tensor
μ	fluid viscosity ($Pa \cdot s$)
μ_{sgs}	turbulent eddy viscosity

ENGINE MODELING SOFTWARE

Engine and combustion modeling encompass most physical processes that can occur in heat, mass and momentum transfer. This complex physics requires as accurate of models and methods we can apply and invent to achieve more predictive modeling. Meeting the accuracy goal is just one of the needs of researchers and industry; they also require fast and robust solution methods that are fairly easy and quick to implement for engine design and analysis. Over the past few years, as outlined above, we have undertaken the task of meeting these goals. Our efforts are proving effective as discussed in this paper and should facilitate the solution of problems encountered in today's engines and those yet to be envisioned.

The FEARCE toolkit provides an excellent platform for developing better in-cylinder fuel and species evolution, including sprays associated with injection, and wall film formulation and evaporation. The code modules and FEARCE in general are a robust and accurate engine simulator, providing quick turn-around required by design engineers. The governing

and model equations are discretized into an FEM form. Physical processes encountered in an engine, often very turbulent reactive and at other times, fairly benign, giving rise to a need to be able to model highly unsteady flow with often very steep gradients.

NAVIER-STOKES EQUATIONS

Fluid momentum is solved for all flow regimes with a Petrov-Galerkin stabilized FEM form. The Petrov-Galerkin modeling of the advection term has 3rd order spatial accuracy.

Turbulent Flow with Multi-Species

The Favre-filtered continuity, momentum, energy and species which govern the evolution of large-scale eddies are expressed as

$$\frac{\partial \bar{\rho}}{\partial t} + \frac{\partial (\bar{\rho} \tilde{u}_i)}{\partial x_i} = 0 \quad (1)$$

$$\frac{\partial (\bar{\rho} \tilde{u}_i)}{\partial t} + \frac{\partial (\bar{\rho} \tilde{u}_i \tilde{u}_j)}{\partial x_j} = \frac{\partial \tilde{\tau}_{ij}}{\partial x_j} - \frac{\partial \bar{p}}{\partial x_i} + \frac{\partial \tau_{ij}}{\partial x_j} + \bar{f}_{drop} + \bar{\rho} \sum_{k=1}^{NumSpecies} \tilde{Y}_k f_{k,i} \quad (2)$$

where $\tilde{\tau}_{ij}$ is the stress tensor evaluated using the Stoke's hypothesis as

$$\tilde{\tau}_{ij} = \mu \left(\frac{\partial \tilde{u}_i}{\partial x_j} + \frac{\partial \tilde{u}_j}{\partial x_i} \right) - \frac{2}{3} \mu \frac{\partial \tilde{u}_k}{\partial x_k} \delta_{ij} \quad (3)$$

$$\begin{aligned} \frac{\partial \bar{E}}{\partial t} = & - \frac{\partial}{\partial x_i} (\tilde{E} \tilde{u}_i + p \tilde{u}_i) + \frac{\partial}{\partial x_i} \kappa \frac{\partial \bar{T}}{\partial x_i} - \frac{\partial (C_p q_i)}{\partial x_i} \\ & + \frac{\partial}{\partial x_i} (t_{ij} + \tau_{ij}) + \frac{\partial}{\partial x_i} (\bar{\rho} \sum_{k=1}^{NumSpecies} \tilde{H}_k (D_k + \frac{\mu_{sgs}}{Sc_t}) \frac{\partial \tilde{Y}_k}{\partial x_i}) \\ & + \bar{\rho} \sum_{j=1}^{NumSpecies} \tilde{Y}_j f_j(x_i) \cdot \tilde{u}_i - \sum_{k=1}^{NumSpecies} H_{o,k} w_k \end{aligned} \quad (4)$$

$$\begin{aligned} \frac{\partial \bar{\rho} \tilde{Y}_f}{\partial t} = & - \frac{\partial}{\partial x_i} (\bar{\rho} \tilde{u}_i \tilde{Y}_f) + \frac{\partial}{\partial x_i} \bar{\rho} \left[\left(D_{f,N} + \frac{\mu_{sgs}}{Sc_t} \right) \frac{\partial \tilde{Y}_f}{\partial x_i} \right] \\ & + \bar{\rho} \tilde{Y}_f f_f(x_i) + \dot{w}_{chem}^f + \dot{w}_{spray}^f \end{aligned} \quad (5)$$

The SGS stress tensor τ_{ij} and SGS heat flux vector q_i in Eqs. (2) and (4) are defined respectively as

$$\tau_{ij} - \frac{1}{3} \tau_{kk} \delta_{ij} = -2 \mu_{sgs} \left(\tilde{S}_{ij} - \frac{1}{3} \tilde{S}_{kk} \delta_{ij} \right) \quad (6)$$

$$q_j = -\frac{\mu_{\text{sgs}}}{\text{Pr}_{\text{sgs}}} \frac{\partial \tilde{T}}{\partial x_j} \quad (7)$$

where μ_{sgs} is the SGS viscosity, Pr_{sgs} is the SGS Prandtl number, and $\tilde{S}_{ij} = \frac{1}{2}(\frac{\partial \tilde{u}_i}{\partial x_j} + \frac{\partial \tilde{u}_j}{\partial x_i})$ is the strain rate tensor. Here variables with \sim are Favre-filtered variables obtained from filtering its grid-filtered component —. In this study, the box or top hat filter is applied for the grid-filtered component.

Mass Conserving Projection Method for Compressible

In order to create a fractional split method, an initial guess for specific momentum is advanced in time, utilizing the expression

$$\Delta U_i^* = -\Delta t \times M_u^{-1} [A_u U_i + K_{\text{tu}} U_i - F_u]^n \quad (8)$$

K_{tu} is the diffusion matrix, A_u is the advection term and F_u is the source term [11-12]. The projection method is presented here with solving momentum explicitly, but the same algorithm is used when our governing equations are solved implicitly.

$$\Delta U_i^* = U_i^* - U_i^n \quad (9)$$

The corrected momentum is determined from the estimated momentum and the pressure gradient, given by

$$U^{n+1} - U^* = -\Delta t \frac{\partial P'}{\partial x_i} \quad (10)$$

Changes of density or pressure are determined from solving an implicit pressure/density Poisson equation created as a result of conservation of mass. This leads to the value for P' as shown in the following continuity solution process:

Mass conservation

$$\frac{\partial \rho}{\partial t} = -\frac{\partial \rho u_i}{\partial x_i} = -\frac{\partial U_i}{\partial x_i} \quad (11)$$

Time advancement in discrete terms for continuity is

$$\frac{\rho^{n+1} - \rho^n}{\Delta t} = -\frac{\partial U_i'}{\partial x_i} \quad (12)$$

where

$$U' = \theta_1 U^{n+1} + (1 - \theta_1) U^n \quad (13)$$

with $U = \rho u$.

We define $P' = \theta_2 P^{n+1} + (1 - \theta_2) * P^n$ and $\Delta P = P^{n+1} - P^n$.

The final specific momentum as obtained with the explicit corrector defined previously by

$$\begin{aligned} \Delta U_i &= U^{n+1} - U^n = \Delta U^* - \Delta t \frac{\partial P'}{\partial x_i} \\ &= \Delta U^* - \Delta t \left(\theta_2 \frac{\partial \Delta P}{\partial x_i} + \frac{\partial P^n}{\partial x_i} \right) \end{aligned} \quad (14)$$

The final specific momentum is obtained using the corrector,

$$\{U_i^{n+1}\} = \{\Delta U_i\} + \{U_i^n\} \quad (15)$$

The mass velocity or momentum is solved and velocity is extracted

$$u^{n+1} = U^{n+1} / \rho^{n+1} \quad (16)$$

Density is recovered from the equation of state. The algorithm for calculating pressure P will be given in VOF section.

TURBULENCE MODELING

A dynamic Vreman type LES method which can transition through laminar to fully turbulent flow is adopted in our work. It requires no assumptions about the turbulent sublayers near walls in bounded flows; this is ideal of engines where the turbulent wall layers are never in equilibrium and the flow is not always very turbulent. A dynamic Vreman model is implemented for modeling turbulent combustion in EMS as described in Waters and Carrington and Waters, et al. [8-12]. Unlike most turbulence models, this VM-LES model does not involve any explicit filtering, averaging, or clipping procedure to stabilize the numerical procedure, enabling it to be used in simulations of reacting flows with complex geometries. For more details regarding the filtering system of this dynamic LES method, please refer to [9]. To date we have investigated flow in benchmark problems assuming all species are air without any chemical reactions; this is a first step to show a better turbulence modeling for EMS combustion software and is part of the larger effort to enhance combustion predictability and efficiencies within engines. As feedback is provided by users, we propose to support the LES modeling in conjunction with reactive chemistry and sprays.

Two-equation $k-\omega$ model Reynolds Averaged Navier-Stokes (RANS) of Wilcox is employed as an alternative to the LES system. This model is more effective than $k-\epsilon$ and as good as or better than $k-\epsilon$ RNG or any other 2-equation model [13]. It handles compressible boundary layers easily, can be integrated to the wall, and is applicable to all flow turbulent regimes. Details of the improved Wilcox's $k-\omega$ are discussed with results to experimental data are found in the papers by Carrington, et. al [12].

Implicit solution method

Developing an implicit solution scheme allows for a larger time step size and maintaining the accuracy of the overall system of model equations. In this system, only advection and source terms are for the load vector. For simplicity, we drop all of the superscripts and source terms.

$$\rho^* \mathbf{u}_i^* - \Delta t \times \left(\frac{\partial \mathbf{t}_i^*}{\partial x_j} - \frac{\partial \mathbf{t}_j^*}{\partial x_i} \right) = \rho^* \mathbf{u}_i^n - \Delta t \times \frac{\partial (\rho^* \mathbf{u}_i^n \mathbf{u}_j^n)}{\partial x_j} \quad (17)$$

Where \mathbf{u}_i^* is the intermediate velocity, \mathbf{t}_{ij}^* is the stress tensor and \mathbf{t}_{ij}^* is the SGS stress tensor. Using the mass conserving projection method described previously results in multiplying \mathbf{u}_i^* by ρ^n to form \mathbf{U}_i^* , hence $\Delta \mathbf{U}_i^* = \mathbf{U}_i^* - \mathbf{U}_i^n$, as usual in the semi-implicit projection.

After determining the pressure as stated earlier, the specific internal energy is solved again in implicit form:

$$\begin{aligned} E^{n+1} - \Delta t \times \frac{\partial}{\partial x_i} \left(\frac{\kappa}{C_v} \frac{\partial E^{n+1}}{\partial x_i} \right) - \Delta t \times \frac{\partial}{\partial x_i} \left(\frac{C_p \mu_{sgs}}{C_v} \frac{\partial E^{n+1}}{\partial x_i} \right) \\ = E^n - \Delta t \times \frac{\partial}{\partial x_i} \left(E^n \mathbf{u}_i^{n+1} + P^{n+1} \mathbf{u}_i^{n+1} \right) + \Delta t \times \frac{\partial}{\partial x_i} \left(\mathbf{t}_{ij}^{n+1} + \mathbf{t}_{ij}^{n+1} \right) \end{aligned} \quad (18)$$

Here, $E^{n+1} = \rho^n e^{n+1}$ and e^{n+1} is the internal energy, and we can get temperature T^{n+1} with $T^{n+1} = \frac{e^{n+1}}{C_v}$.

The solution to the species transport equations follows in a like manner:

$$\begin{aligned} \rho^n \Upsilon_j^{n+1} - \Delta t \times \frac{\partial}{\partial x_i} \rho^n \left[\left(D_{j,N} + \frac{\mu_{sgs}}{Sc_i} \right) \frac{\partial \Upsilon_j^{n+1}}{\partial x_i} \right] = \\ \rho^n \Upsilon_j^n - \Delta t \times \frac{\partial}{\partial x_i} (\rho^n \mathbf{u}_i^{n+1} \Upsilon_j^n) \end{aligned} \quad (19)$$

Our implicit system is not fully implicit because we want to keep the matrix as symmetric as possible for large-scale parallel calculation to reduce computational time such as for preconditioning. By using the implicit method, the solution speed is increased by about 10x although the time step is on the order of 100 to 1000 times larger. These results are shown in the paper by Waters and Carrington [7].

The solution of the species transport equations are generally more troublesome than the fluid's momentum equation. Hence, 1st order accurate will be problematic, with the solver producing somewhat incorrect momentum on a resolved grid, and be wholly inaccurate with the species transport. One does not necessarily mean the other is correct from momentum to species. Our system is 3rd order accurate on all advection terms, assuring greater accuracy in addition to the greater accuracy per cell of the same size as other more traditional methods. The key to combustion modeling lies in having the correct species transport, the species concentration in the right place at the right time. Our EMS code with its inherent accuracy facilitates this solution.

All flows are handled in our one code, from subsonic to hypersonic, and from laminar to fully turbulent. The LES handles the transitional flow from laminar to fully turbulent flow found in engine fluids/gases, as the system is highly unsteady.

SPRAY, PARTICLES AND INJECTION MODELING

Another key component of combustion modeling for engines is fuel injection. Modeling of the injected fuel liquid stream as it initially breaks into ligaments and subsequently becomes atomized further downstream is fundamental to eventual distribution of fuel.

Engineering models for Spray

A KH-RT model, and a TAB model in FEM inclusive of: Collision, Evaporation, Wall filming Processes, Droplet Splashing. More information about those process can be found in [17-19]. A soot model (3 soot particle types) is included in EMS as part of the particle tracking system.

KH-RT spray modeling

A Kelvin-Helmholtz and Rayleigh-Taylor spray model is designed to simulation from early stage injection to late stage ligament and droplet atomization. Injection is simulated with large drop parcels, having only one particle in each parcel. This mimics the liquid core at entry into the combustion chamber. This is similar to the blob injection model by Rietz [20] with a change being only one particle in the parcel at point of injection. The details of the entire modeling process in the KH-RT can be found in the monograph (with references to various components) by Stiesch [21]

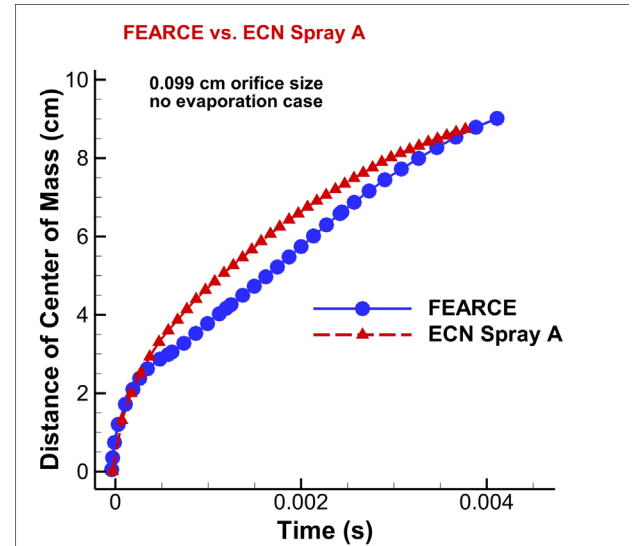


Fig. 1: Solution for an ECN Spray A case compared with experimental data

Shown in Fig. 1 are results of the implement KH-RT system for the Spray A (diesel) case from the Engine Combustion Network [22]. The nozzle is 0.09cm with ambient temperature of 435K. Shown is the result compared to experiment on the coarsest grid resolution of 4mm. The solution compares favorably to experimental data.

Predictive Spray Break-Up

Two-Phase Flow is solved for the liquid Being Injected into the engine. This is a true two-phase modeling capability, using a

Volume of Fluids (VOF) method [23-25]. Secondary droplet transport from Predictive primary break-up is in development.

- **VOF equation:**

$$\frac{\partial f}{\partial t} + \mathbf{U} \cdot \nabla f = 0 \quad (20)$$

where f is the volume of fraction.

- **Momentum equations** (FEM Projection method):

If $f = 1$, run this incompressible equation for u_i^* , which is the intermediate velocity for predictor-corrector split projection method:

$$\frac{\rho^n u_i^* - \rho^n u_i^n}{\Delta t} = -\rho^n \mathbf{U} \cdot \nabla u_i + \left(\mu + u_{sgs} \right) \frac{\partial^2 u_i}{\partial x_j^2} + \delta \kappa n \delta_\Gamma \quad (21)$$

If $f \neq 1$, run this compressible equation for u_i^* :

$$\frac{\rho^n u_i^* - \rho^n u_i^n}{\Delta t} = -\rho^n \mathbf{U} \cdot \nabla u_i + \frac{\partial t_{ij}}{\partial x_j} + \frac{\partial \tau_{ij}}{\partial x_j} + \delta \kappa n \delta_\Gamma \quad (22)$$

where t_{ij} is the stress tensor, and τ_{ij} is the SGS stress tensor, \mathbf{U} is the velocity vector. δ is the surface tension coefficient, κ is the surface curvature n is the unit normal of the interface surface and δ_Γ is a Dirac Delta function.

- **Pressure Solve:**

Let $\Delta u^* = \rho^n u_i^* - \rho^n u_i^n$ and $\Delta P = P^{n+1} - P^n$. P^n is the pressure at time step n . Since $\rho^n u_i^{n+1} - \rho^n u_i^* = -\Delta t \frac{\partial P'}{\partial x_i}$, where $P' = \theta_2 P^{n+1} + (1 - \theta_2) P^n$ and $\frac{1}{c^2} \Delta P = \Delta \rho = -\Delta t \frac{\partial \rho^n u_i^*}{\partial x_i}$, where $\rho^n u_i^* = \theta_1 \rho^n u_i^{n+1} + (1 - \theta_1) \rho^n u_i^n$, then we have

$$\frac{1}{c^2} \Delta P - dt^2 \theta_1 \theta_2 \frac{\partial^2 P}{\partial x_i^2} = dt^2 \theta_1 \frac{\partial^2 P'}{\partial x_i^2} - dt \left(\theta_1 \frac{\partial \Delta u^*}{\partial x_i} + \frac{\partial \rho^n u_i^*}{\partial x_i} \right) \quad (23)$$

Therefore $P^{n+1} = P^n + \Delta P$, c is the sound speed. When it is compressible flow, sound speed is calculated by $c = \sqrt{\gamma R T}$, where R is the specific gas constant and T is the temperature. When it is incompressible flow, we use artificial compressibility β for c .

$$\beta = \max(\epsilon, u_{conv}, u_{diff}) \quad (24)$$

where ϵ is a small constant to ensure β is not approaching zero. $u_{conv} = |U| = \sqrt{u_i u_i}$ and $u_{diff} = \frac{v}{h}$, where h is the element size and v is the kinematic viscosity. Refer to Zienkiewicz [26] for more details about artificial compressibility. In order to balance the pressure at region $0 < f(x) < 1$, we take the control volume integral $P = \frac{\int_e P dV}{V_e}$ for every point in that area to avoid the pressure being discontinuous, where e is all of the elements associated with that point and V_e is the volume of all of those elements related to that point.

- **Velocity Update:**

Update the velocity u_i^{n+1} as described in Equations (14-15).

- **Energy equations:**

If $f > 0$, run the incompressible equations for internal energy E_{liquid} . If $f < 1$, run the compressible equations for internal energy E_{gas} . Then the internal energy is calculated as:

$$E = f \times E_{liquid} + (1 - f) \times E_{gas} \quad (25)$$

- **Species equations:**

Define mass fraction as $Y_j = \frac{\rho_j}{\rho_{gas}}$, where ρ_{gas} is the gas density and ρ_j is species j . Here we only consider the gas species for mass fraction. The liquid density is tracked according to VOF.

- **Aggregation for gas properties**

Use Y_j and its species properties to aggregate the gas properties for $C_{p,gas}$, $C_{v,gas}$, κ_{gas} and μ_{gas} , the same procedure by Carrington [11]. If $f < 1$, $\rho_{gas} = \frac{P}{RT}$. Now we can aggregate the properties by VOF

$$\mu = f \times \mu_{liquid} + (1 - f) \times \mu_{gas} \quad (26)$$

$$\rho = f \times \rho_{liquid} + (1 - f) \times \rho_{gas} \quad (27)$$

$$\text{Temperature } T_{liquid} = \frac{E_{liquid}}{C_{v,liquid}} \text{ for } f > 0 \text{ and } T_{gas} = \frac{E_{gas}}{C_{v,gas}} \text{ for } f < 1$$

. Then

$$T = f \times T_{liquid} + (1 - f) \times T_{gas} \quad (28)$$

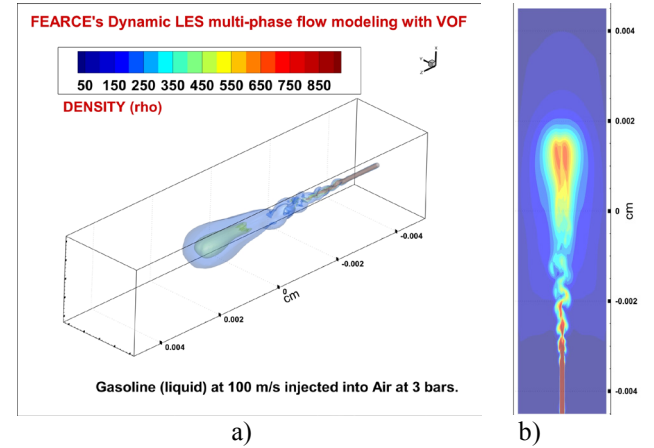


Fig. 2 Primary spray break-up using VOF (represented by density (ρ)), a) 3-D view, b) 2-D slice meridional plane.

Figure 2 show a liquid jet breaking apart from shear stress between the gas and liquid surface. The primary spray break-up into ligaments compares well to the DNS solutions as described in the paper by Waters, Carrington and Francois [23].

HIGH ACCURACY WITH AN ADAPTIVE FEM

Incorporating h -adaptive or grid refinement FEM with moving parts system, provides high computational accuracy and yield better converge rates. The use of h -adaptation [9-11] refines the mesh where the local relative error is large (measured by a percentage of the average total error in the domain). This refinement process assists LES modeling by producing a solution with a specified error on the domain utilizing a minimal number of elements, thereby reducing the computational time, i.e., minimizes the computer time of solution for a given error in the solution. In addition, this h -adaptive method is especially helpful to capture shocks and other flow features that might not be resolvable with the grid resolution used at the start of a simulation. Adaptive FEM technique is a powerful tool in CFD modeling. Since the pioneer work done by Peraire *et al.* [27] in 1987 for accurately capturing shock waves by using h -adaptive FEM technology, adaptive FEM techniques have been widely used in various areas [28-34].

The hp -adaptive FEM strategy employed in this research follows a three-step strategy and is guided by a posterior error estimator based on the L_2 norm. Three consecutive hp - adaptive meshes are constructed for solving the system in order to reach a preset target error- an initial coarse mesh, an intermediate h -adaptive mesh, and a final hp - adaptive mesh obtained by applying p - adaptive enrichments on the intermediate mesh. The p - adaptation is continued when the problem solution is pre-asymptotic. Both global error and local error conditions have to be met for an acceptable solution [31].

Among the adaptation family, the hp -adaptive FEM is one of the best mesh based algorithms. The computational mesh is automatically refined and unrefined based on the of a stress error measure, or a residual or gradient method; the grid and shape function order is dynamically controlled by the computational error [33].

The computational mesh is automatically refined and unrefined based on the change of flow features, see Pepper and Carrington [34]. The smooth flow region usually associates with small computational error, and the fast changing flow region usually associated with large computational error, as discussed in Waters *et al.* [9]. Mesh adaptation starts with an initial coarse mesh, then develops as a local element refinement indicator is used define if a local refinement for an element is needed, i.e.

$$\xi_i = \frac{\|e\|_i}{\bar{e}_{avg}} \quad (29)$$

When $\xi_i > 1$, the element is refined; when $\xi_i < 1$, the element is unrefined. The local relative percentage error of any single element is $\|e\|_i$ while the average element error is defined as \bar{e}_{avg} . The gradient of the speed in each element is used to calculate the error estimate. There are different levels of refinement. When an element is refined once this is labeled a level one refinement, refining again produces a level two for refinement, and so on. Virtual nodes are interpolated or the matrix is reduced to preserve

flux balancing. More details can be found in Wang and Pepper [30].

IMMERSED MOVING BOUNDARIES

Immersed Boundary Method (IBM) for unstructured meshes models the moving parts in the engine simulation. Reactive flow in engines has highly variable conditions in and around the combustion chamber, where moving parts provide for mass efflux and substantial changes to fluid and thermodynamic states, creating a difficult task to solve. This is made more complicated by the need to preserve spatial accuracy of interface all associated physics contributed by the moving parts. These are moving boundaries that require extreme care in modeling if we are to seek a predictive modeling capability, one that needs to be as exact as possible, precisely locating the parts, not inducing numerical dispersion into the system being solved, and exactly providing the boundary conditions for the flow.

We have addressed this issue with the development of a both a local-ALE method and an Immersed Boundary Method (IBM) for unstructured meshes that remain fixed during the calculation, while the moving devices are allowed to slide independently over the mesh [14-16]. The IBM method employs moving markers on the surfaces as developed in the local-ALE scheme. This approach has the advantage that it eliminates the need for re-meshing and interpolating between meshes. Only one single non-body-conformal mesh is used throughout the entire simulation, in addition, there is no limit to the number moving parts in the system, provided they don't collide. The issues related to the local generation of badly shaped elements occur at the local mesh size scale and their associated error is effectively removed, such as dealing with "small-cell" instability problem occurred in cut cell method. Also it only solves active fluid nodes, hence reducing the computational time significantly. Furthermore, it is easier for the implementation of h -adaptation. The use of "immersed" nodes and "ghost" nodes ensure the boundary conditions are satisfied precise on the immersed boundary. Shown in Fig. 3, if the cell is not being cut, and it is on the fluid side, all nodes in this cell are marked as fluid nodes. Similarly, if the cell is not being cut and it is on the solid side, then all nodes in this cell are labeled as solid nodes. If the cell is being cut by a surface, then the nodes on the fluid part will be "immersed" nodes while the nodes on the solid part will be "ghost" nodes. Except for the solid nodes, all other nodes are active nodes.

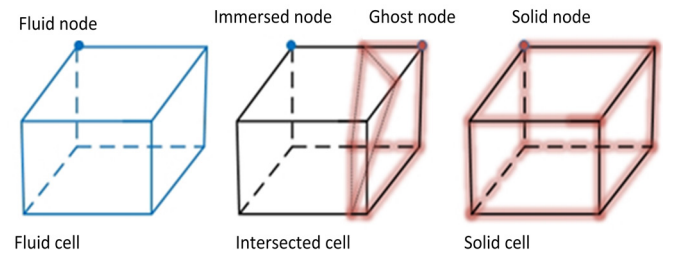


Fig. 3 Declaration of nodes for IBM

The interpolation process can be done with any order of interpolation. In order to ensure mass conserves, the same order of interpolation has to be applied for all variables, see [38]. In our work, we used just linear interpolation. For “immersed” node, we interpolated with the closest fluid node along the normal direction of the immersed boundary surface and the boundary value of the immersed boundary. This is only done for velocity and temperature. For “Ghost” node, we take the average value of the fluid nodes that share the same element as the “Ghost” node. This is only for pressure and species. Find the closet fluid node and surface node along the normal direction and Linearly extrapolate ghost node from those velocity and temperature (for variables that have fixed values on the immersed boundary). This ensures mass conserve.

The method has been implemented in EMS using unstructured meshes and tested on problems with several moving objects, curved piston surfaces, 4 valve engines, etc...

GRID GENERATION

Simplistic initial grid generation yet producing quality hexahedral elements. Only the convex domain requires gridding; there is no concern with taken for the moving parts such as valves, pistons or vanes. The moving parts represented as surface files of stereolithographic type form overlay the convex gridded domain. The IBM method takes care of the motion through the Eulerian grid. No remeshing, no mesh tangling, good elements for integration are therefore provided.

Initial grid generation simplicity for quality hexahedral elements is key to reducing labor cost and allow for quicker turn-around times in the engine design process. We have developed the use of GridPro and Cubit grid generation using the overset grid for moving parts as a start in the process. We have found the grids from GridPro to be more ideally suited for fluid dynamics, with quality hex elements. We developed our overset grid system to be used in conjunction with the GridPro but can be used with any block-structured or unstructured grid generation software. We have developed convertors from both Cubit and GridPro outputs to the EMS system that is easy and simple to use, quickly making for the data and grid decks and moving parts marker system.

The grid in Fig. 4 was developed to model one of Sandia’s Combustion Research Facility test engines, a Direct Injected Spark Ignition (DISI) engine. The grid is generated automatically from what is known as topological representation of the engine. That topology wraps or maps the surfaces of the engine (surfaces can come from CAD) and provides the foundation to the gridding process. This topology is inherently fairly genetic, and as such can represent many different engines configurations; all that would need be done on various engine configurations is to reassign the surfaces and perhaps make modest adjustments to the topology. With this system shown in Fig. 4, the moving parts are overlaid onto the grid, and algorithms in EMS manage the movement through the underlying hexahedral grid. This gridding system is

straightforward as gridding goes, easy for a quality grid generator to perform.

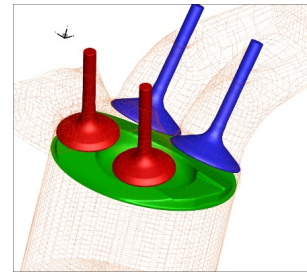


Fig. 4 FEARCE DISI test case composed of hexahedral grid with our overlaying parts for IBM method

HIGH SPEED SOLUTION METHODS

Computer code using parallel methods including MPI integrated into the adaptive modules, Algorithm developments for faster solutions, Super-linear scaling, must faster than previous LANL versions of engine modeling software [2-3]. The FEEARS code either uses a semi-explicit or an implicit solver system, both of which use the MPI paradigm for parallel communication, and truly a requirement for 3D simulations. For the implicit solution processes, the PCG linear equation solver package having various Krylov solvers with user supplied matrix-vector multiplication and dot product operations is incorporated. In addition, the PCG package provides for user developed preconditioning methods and a global system for convergence at each time iteration. In this case for the global convergence system, we use a Beam-Warming method for evoking the solution to the equations that uses an additive Schwartz preconditioning system. Communication for preconditioning requires the matrix-vector multiple and vector dot product are user supplied which require collocations, that is, gather and reduction processes from MPI facilitated by utilizing the array of globally shared nodes stored on mother processor. The Beam-Warming iteration process provides for the global convergence of the equation system as developed and described by Carrington [36]. For the semi-explicit process, the same PCG linear equation solver is used for the pressure and for the implicit method, it is also used for all the equations (although it is recommended to subsycle the species transport equation since there are a great many replications of this for each species and is therefore easily threaded onto processing units). Details regarding MPI implementation and efficiency are given in Waters and Carrington [7] and show nearly 30 times speed-up for strong scaling over the serial implementation of the overall CFD solver.

REACTIVE CHEMISTRY

ChemKin-Pro and ChemKin II have been implemented to support many in industry and many researchers use ChemKin. The ANSYS ChemKin-Pro is fully supported software that reflects newest developments in the reaction chemistry software [37]. The EMS code has interfaces to ChemKin-II and ChemKin-Pro. Many in industry and many researchers use

ChemKin-Pro and many other researchers use ChemKin II or III. The ANSYS ChemKin-Pro is fully supported software that reflects newest developments in the reaction chemistry software. For the sake of brevity we refer the reader to references where the information can be found for the reactive chemistry packages [37].

MOTORED ENGINE SIMULATION

The simulated engine is the optically accessible (DISI) engine at Sandia National Laboratories. This engine shown in Fig. 4 has 4 valves, 2 intake valves (blue) and 2 exhaust valves (red). The optical engine geometric specifications and engine operating conditions are given in [38]. Here we only showed case 0 for engine operating conditions. In this study, a vertical mesh partition is implemented in light of the fact that cells can become deactivated and activated during a simulation, which will affect the load balance between processors. Cells are partitioned according to the direction of piston motion will optimize the computational time significantly [39]. In our work, we use ParMETIS, a software package for parallel partitioning unstructured graphs. Figure 5(a) shows the vertical partitioning of the entire domain with 12 processors. Comparing with all equal weights partition Fig. 5(b), the computational time is reduced by half. Figure 6 is showing the velocity vectors at different crank angle (CA), where we can see the fluid is well mixed at 192 CA. The valves will be closed at 226 CA and the compression will start. The averaged in-cylinder pressure is around 100.74KPa at 225.52 CA in Fig. 7(a) right before the valves are fully closed. At 360 CA it reaches the highest pressure which is 2800KPa in Fig. 7(b) and it is higher than the data in [38] due to the exact mass intake being calculated not matching the exact in the experiment. Also the averaged flow-field vectors and streamlines are plotted in Fig. 8 at 335CA which is close to the spark timings for the selected fired operating conditions. From Fig. 8(b), we see the highest velocity is about 6m/s which matches the experimental data [38]. Our LES gave a more turbulent flow than [38] because we didn't use the Perturbation Methodology to get the mean flow field. However, we are able to capture a center rotation in this vector slice taken from 9mm from the pent roof Fig. 8(a). A further thorough analysis will be given in our next work.

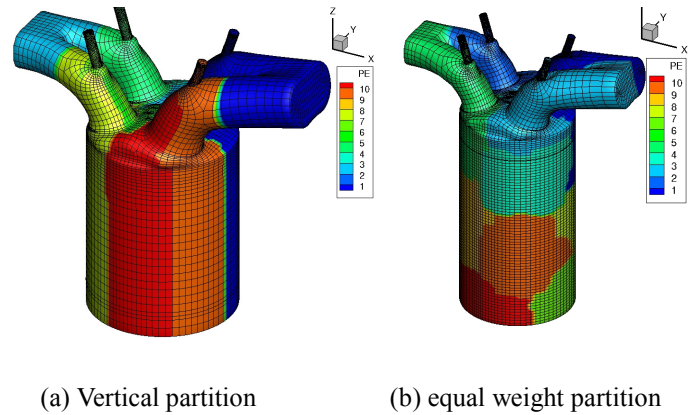


Fig. 5 Different partitioning. PE is the number of processors.

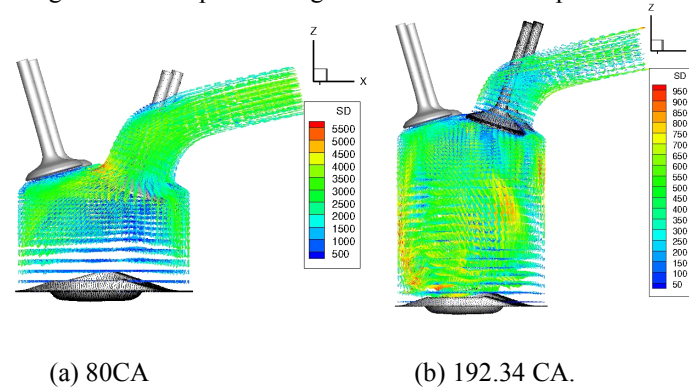


Fig. 6 Velocity vectors at different crank angle(CA)

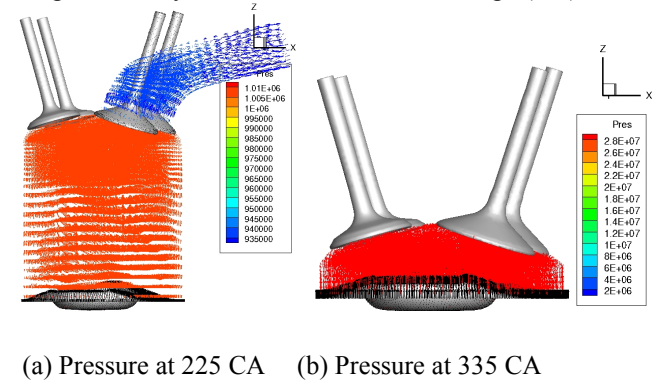


Fig. 7 Pressure and turbulent viscosity at different CA.

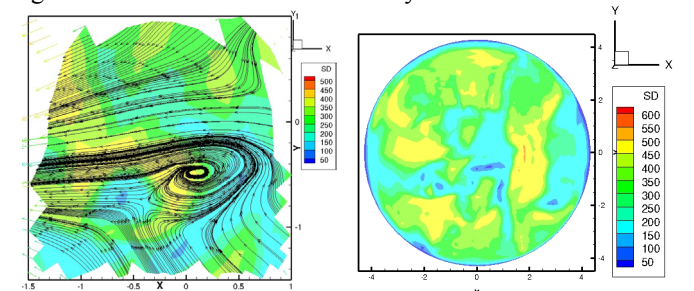


Fig. 8 Gas flow vector and streamlines at nominal spark timing 335 CA.

CONCLUSION

This work is to demonstrate the capability of our software FEARCE toolkit to model fluid mechanicals, more importantly simulating engines. We gave our fundamental numerical algorithm, which is finite element method with mass conserving projection method. FEARCE runs under either explicit or implicit scheme. It has both dynamic LES and Two-equation $k-\omega$ model (RANS) for modeling turbulent flow. The theory for both techniques is given in this study. Spray, particles and injection are handled by VOF and KH-RT break up modeling. Here we illustrated how well our Spray, particles and injection models can be. IBM is implemented to take care of all moving parts, such as valves and pistons. IBM has more advantage over other existing method modeling moving parts e.g. cut cells and ALE. The entire code is parallelized by MPI and it can achieve significant speed up. Last but not least, the simplicity of our mesh generation will make this software easier to use. In this work we only gave a brief overview of our FEARCE toolkit's capability modeling realistic engines. More detailed analysis and theory will be given in the future work.

ACKNOWLEDGMENTS

The DOE's Office of Energy Efficiency and Renewable Energy (EERE) Advanced Combustion Program (Gurpreet Singh) is supporting this effort. Los Alamos National Laboratory, an affirmative action/equal opportunity employer, is operated by the Los Alamos National Security, LLC for the National Nuclear Security Administration of the U.S. Department of Energy (DOE) under contract DE-AC52-06NA25396. Los Alamos National Laboratory strongly supports academic freedom and a researcher's right to publish; as an institution, however, the Laboratory does not endorse the viewpoint of a publication or guarantee its technical correctness.

REFERENCES

- Desjardins P. E. and Frankel S. H., Two-dimensional Large Eddy Simulation of soot formation in the near field of a strongly radiating non-premixed acetylene-air jet flame, *Combust. Flame*, 119, 121-133, 1999.
- Colin O., Ducros F., Veynante D. and Poinot T., "A thickened flame model for large eddy simulations of turbulent premixed combustion", *Phys. Fluids*, 12, 1843-1863, 2000.
- Pierce C. D. and Moin P., "Progress-variable approach for large eddy simulation of non-premixed turbulent combustion", *J. Fluid Mech.*, 504, 73-97, 2004.
- Selle L., Lartigue G., Poinot T., Koch R., Schildmacher K.-U., Krebs W., Prade B., Kaufmann P. and Veynante D., Compressible Large-Eddy Simulation of turbulent combustion in complex geometry on unstructured meshes, *Combust. Flame*, 137, 489-505, 2004.
- Waters, J., Carrington, D.B., Pepper, D.W., "An Adaptive Finite Element Technique with Dynamic LES for Incompressible and Compressible Flows, *Proceedings of the 15th Computational Heat Transfer Conference, CHT-15*, Piscataway, New Jersey, May 25-29, 2015.
- Waters J., Carrington, D.B., "Parallel Large Eddy Simulation for Modeling 3D Turbulent Flow in Engines", *Proceedings of the ASME 2016 Fluids Engineering Division Summer Meeting*, July 10-14, 2016, Washington, DC, USA
- Waters J., Carrington, D.B., "A Parallel Large Eddy Simulation in a Finite Element Projection Method for all Flow Regimes", *Numerical Heat Transfer, Part B: Fundamentals* 72:4, pages 285-299. 2017.
- Waters J., Carrington, D.B., Pepper, D.W., "Parallel Large Eddy Simulation for Turbulent Reactive Flow Modeling," *Procs. of the Int. Conf. Computational & Experimental Engr. & Sci. (ICCES'15)*, Reno, NV., 2015.
- Waters J., Carrington, D.B., Pepper, D.W., "An Adaptive Finite Element Method with Dynamic LES for Turbulent Reactive Flows", *Computational Thermal Sciences*, 8 (1): 57-71, 2016.
- Waters J., Carrington, D.B., Pepper, D.W., "Application of a dynamic LES model with an H-adaptive FEM for fluid and thermal processes," *Procs. of 1st Thermal and Fluid Engineering Summer Conference - TFESC*, 2015-08-09/2015-08-12, N.Y., N. Y., United States, (2015)
- Carrington, D. B. , "A Fractional step *hp*-adaptive finite element method for turbulent reactive flow," Los Alamos National Laboratory Report, LA-UR-11-00466, 2011
- Carrington, D.B., Wang X., Pepper D.W., "A predictor-corrector split projection method for turbulent reactive flow," *Comput. Therm. Sci*, Begell House Inc., v5, #4, pp.333-352, 2014
- Menter, F.R., 1992, Improved Two-Equation $k-\omega$ Turbulence Models for Aerodynamic Flows, NASA Report no. TM-103975
- Carrington, D.B., Munoz, D., Heinrich J.C., "A local ALE for flow calculations in physical domains containing moving interfaces," *Progress in Computational Fluid Dynamics*, an Int. Jour. vol 14, no, 3, pp. 139-150, 2014
- Hatamipour, V.D., Carrington, D.B., Heinrich, J.C., "Accuracy and Convergence of Arbitrary Lagrangian-Eulerian Finite Element Simulations based on a Fixed Mesh," *Progress in Computational Fluid Dynamics*, an Int. Jour., vol. 18, no 4, pp. 215-231, 2018
- Carrington, D.B., Mazumder, M., Heinrich, J.C., "Three-Dimensional Local ALE-FEM Method for Fluid Flow in Domains Containing Moving Boundaries/Objects Interfaces," *Progress in Computational Fluid Dynamics*, an Int. Jour., vol. 18, no 4, pp. 199-215, 2018
- Dukowicz, J.K., "A Particle-Fluid Numerical Model for Liquid Sprays," *Journal of Computational Physics*, vol. 35, 2, April 1980
- Amsden, A. A., O'Rourke, P. J., Butler, T. D., "KIVA-II, a computer program for chemically reactive flows with sprays," Los Alamos, N.M. : Los Alamos National Laboratory Scientific Report, LA-11560-MS, 1989.

19. Torres, D.J., O'Rourke, P.J., Amsden, A.A., "Efficient multicomponent fuel algorithm," *Combustion Theory and Modelling*, vol.7, 1 2003, p.66-86
20. Reitz, R.D., "Modeling Atomization and Processes in High-Pressure Vaporizing Sprays," *Atomization and Sprays Technology*, vol 3, 1987, pp. 309-337.
21. Stiesch, G., "Modeling Engine Spray and Combustion Processes", Springer-Verlag, Berlin, Germany, 2003, pp.130-161.
22. Engine Combustion Network, Spray A, 435 K ambient temperature, 2.93 MPa pressure vessel. [<https://ecn.sandia.gov/ecn-data-search/?nam=1>].
23. Waters, J., Carrington, D.B, Francois, M.M., "Modeling Multi-Phase Flow: Spray Break-up Using Volume of Fluids in a Dynamic LES FEM method," *Numerical Heat Transfer, Part B*, vol. 72, no. 4, pp. 285-299, 2017
24. Grosshans, H., Szasz, R.Z., Fuchs, L., "Full Spray Simulation – Coupled Volume of Fluid and Lagrangian Particle Tracking methods," *ILASS-Europe 2011, 24th European Conference on Liquid Atomization and Spray System*, 2011
25. Grosshans, H., Szasz, R.Z., Fuchs, L., Development of a Combined VOF-LPT Method to Simulate Two-Phase Flows in Various Regimes, *Seventh International Symposium on Turbulence and Shear Flow Phenomena (TSFP-7)*, July 28-31, Ottawa, CA, 2011
26. Zienkiewicz, O.C., Taylor, R. and Nithiarasu, P, "The Finite Element Method for Fluid Dynamics, 6th Edition", Imprint: Butterworth-Heinemann, 2005.
27. Peraire, J., Vahdati, M., Morgan, K. and O. C. Zienkiewicz, 1987, Adaptive Remeshing for Compressible Flows, *Journal of Computational Physics*, Vol., 72, No. 2, pp.449-466.
28. Carrington, D.B., Wang X., Pepper D.W., "An *hp*-adaptive Predictor-Corrector Split Projection Method for Turbulent Compressible Flow, *Proceedings of the 15th International Heat Transfer Conference, IHTC-15*, Kyoto, Japan, August 10-15, 2014.
29. Ilinca, F., Pelletier, D., Garon, A., 1997, An Adaptive Finite Element Method for A Two Equation Turbulence Model in Wall-Bounded Flows, *International Journal of Numerical Methods in Fluids*, Vol. 24, pp. 101 – 120.
30. Nithiarasu, P. and O. C. Zienkiewicz, 2000, Adaptive Mesh Generation for Fluid Mechanics Problems", *International Journal of Numerical Methods in Engineering*, Vol. 47, pp. 629-662.
31. Zienkiewicz, O. C. and R. J. Z. Zhu, 1987, A Simple Error Estimator and Adaptive Procedure for Practical Engineering Analysis, *International Journal of Numerical Methods in Engineering*, Vol. 24, pp. 337-357.
32. Demkowicz, L. 2006, Computing with *hp*-Adaptive Finite Elements. One- and Two- Dimensional Elliptic and Maxwell Problems, Chapman & Hall/ CRC Press, Taylor and Francis.
33. Wang, X. and D. W. Pepper, 2007, Application of an *hp*-adaptive FEM for solving thermal flow problems, *AIAA Journal of Thermophysics Heat Transfer*, Vol., 21, pp.190-198.
34. Pepper, D. W. and Carrington, D. B., 1999, Application of *h*-adaptation for Environmental Fluid Flow and Species Transport, *International Journal of Numerical Methods in Fluids*, Vol. 31, pp. 275 – 283.
35. Kumar M., Roy S., "A sharp interface immersed boundary method for moving geometries with mass conservation and smooth pressure variation," *Computers & Fluids*, Volume' 137, Pages 15-35, 2016
36. Carrington, D., A Parallel First-Order Spherical Harmonics (P1) Matrix-Free Method for Radiative Transport", *Numerical Heat Transfer. Part B: Fundamentals*. 97-117. 2008.
37. ANSYS ChemKin-Pro: <https://www.ansys.com/products/fluids/ansys-chemkin-pro>
38. Van Dam N., Zeng W., Sjöberg M. and Som S., "Parallel Multi-Cycle LES of an Optical Pent-Roof DISI Engine Under Motored Operating Conditions," ASME. Internal Combustion Engine Division Fall Technical Conference, Volume 2: Emissions Control Systems; Instrumentation, Controls, and Hybrids; Numerical Simulation; Engine Design and Mechanical Development ICEF 2017-3603.
39. Torres D., Li Y., Kong S., "Partitioning strategies for parallel KIVA-4 engine simulations," *Computers & Fluids*, Volume 39, Issue 2, Pages 301-309, 2010.

ICEF2016-9327

Modeling Turbulent Reactive Flow in Internal Combustion Engines with an LES in a semi-implicit/explicit Finite Element Projection Method

Jiajia Waters

Los Alamos National Laboratory
Los Alamos, NM, U.S.A.

David B. Carrington

Los Alamos National Laboratory
Los Alamos, NM, U.S.A.

ABSTRACT

A Finite Element Method (FEM) for the solution of turbulent reactive flow on parallel machines is being developed for engine and combustion modeling. This FEM solver is parallel (MPI), solves multi-species fluids using either Reynolds Averaged Navier-Stokes (RANS) $k-\epsilon$ model or dynamic LES model for turbulent reactive flow. The code provides an excellent platform for developing better in-cylinder fuel and species evolution, including sprays associated with injection. This code is more robust and more accurate than current engine simulators, with quick turn-around times required by design engineers. We've also developed a dynamic LES method which can transition through laminar to fully turbulent flow, and hence requires no assumptions about the turbulent sublayers near walls in bounded flows; this is ideal for engines where the turbulent wall layers are never in equilibrium and the flow is not always turbulent. Another key component of combustion modeling for engines is fuel injection. Spray modeling from KIVA is adopted in our work with the ability to solve large 3D turbulent flow by the help of the parallel technique.

INTRODUCTION

Engine combustion involves a number of complex, closely coupled physical and chemical processes. These include the transient three-dimensional (3D) dynamics of evaporating fuel sprays interacting with multicomponent gases undergoing mixing, ignition, chemical reactions and heat transfers along with turbulent flow. Coupling between these processes occurs over a wide range of time and length scales. Further complications arise when multiple phases are present due to the introduction of dynamically evolving interface boundaries and the complex exchange processes that occur as a consequence. Our work has the ability to calculate such flows in engine cylinders, including the effects of turbulence.

The most widely used approach for modeling turbulent flow, Reynolds-Averaged Navier-Stokes (RANS) models, is implemented in the code and is the two-equation $k-\epsilon$ model. Time averaging methods produce a mean value for turbulent

variables, kinetic energy, dissipation rate, and turbulent viscosity and are not able to capture detailed flow structures of unsteady turbulence. Large Eddy Simulation (LES) does provide solution to unsteady turbulence intensity, viscosity, and structures and is becoming widely used to study combustion in many modern combustion devices [1-5]. An LES model solves the spatially averaged Navier-Stokes equations using a filtering process that is based on the grid size. In LES, the larger eddies are directly resolved at the grid resolution and eddies smaller than the grid are modeled. Using a dynamic Vreman model [6] as implemented in the Finite Element Method (FEM) not only guarantees vanishing subgrid-scale (SGS) dissipation for various laminar shear flows, but also eliminates the need to use a wall-damping function in simulations of boundary layer flows, thereby implying that it is a suitable LES system for wall-bounded shear flows [7]. The dynamic Vreman LES model can model various flow regimes, including laminar, transitional and turbulent flows simultaneously. More details of the LES implementation can be found in Waters et al. [8]

In modern engine fuel is most often injected into the combustion chamber. We model this spray injection with a Lagrangian particle transport, the standard KIVA multi-component model [9-11]. The modeling includes two-way coupling process between the spray and the fluid and also turbulence dispersion (as calculated by RANS ($k-\epsilon$) or LES). Los Alamos KIVA Multi-component Spray algorithms by P.J. O'Rourke, Tony Amsden, David J. Torres and John K. Dukowicz [9-12] is adopted in this new engine modeling code to simulate spray dynamics. The Lagrangian particle transport model is a statistical representation of the spray and accounts for a spectrum of droplet sizes, the effects of evaporation, agglomeration, and droplet break-up.

An iterative two-way coupling between fluid and droplets is employed, which usually only takes 2 iterations to converge. The particle transport scheme employs a fast ray-tracing method for associating fluid grid elements with droplet parcels. The FEM method allows for a measure of fluid and thermal properties at each droplet parcel location. Variables utilized by spray

dynamics are exactly (minimum of 2nd order-in-space) represented at grid resolution, so even using a coarse grid resolution produces spatially convergent spray simulation, and is something the usual piece-wise finite volume methods cannot achieve.

Comparing to RANS simulations, LES models require finer mesh because the modeled flow size depends on the filter size which is chosen to be the mesh size, and unstructured CFD algorithms which is common to engine combustion modeling require large computing resources that potentially can be provided by the emerging parallel computer systems. By linking together hundreds or even thousands of individual processor nodes, the parallel computer systems can deliver significant advances in computational resources in terms of memory, storage, and computing speed. Therefore a Message Passage Interface (MPI) is developed in our work for this FEM system to model 3D turbulent flow in engines. Jimack, P. K [13] gave a brief introduction about using MPI for FEM. We developed a parallel method for finite elements for a domain decomposition using a nodal type graph, and producing overlapping or interface elements where the partition dissects the element, leaving some element nodes on another processor. This parallel system also tracks the move of the parcels from one position to another position and one processor to another processor that occurred in the spray model and has the ability to convert the properties from the parcels to fluids properties and vice versa. For the implicit solution processes for pressure the PCG linear equation solver package having various Krylov solvers with user supplied matrix-vector multiplication (matvec) and dot product operations is incorporated.

GOVERNING EQUATIONS

Turbulent Flow with Multi-Species

The Favre-filtered continuity, momentum, energy and species which govern the evolution of large-scale eddies are expressed as

$$\frac{\partial \bar{\rho}}{\partial t} + \frac{\partial (\bar{\rho} \tilde{u}_i)}{\partial x_i} = 0 \quad (1)$$

$$\begin{aligned} \frac{\partial (\bar{\rho} \tilde{u}_i)}{\partial t} + \frac{\partial (\bar{\rho} \tilde{u}_i \tilde{u}_j)}{\partial x_j} &= \frac{\partial \tilde{\tau}_{ij}}{\partial x_j} - \frac{\partial \bar{p}}{\partial x_i} + \frac{\partial \tau_{ji}}{\partial x_j} \\ &+ \bar{f}_{drop} + \bar{\rho} \sum_{k=1}^{NumSpecies} \tilde{Y}_k f_{k,j} \end{aligned} \quad (2)$$

where $\tilde{\tau}_{ij}$ is the stress tensor evaluated using the Stoke's hypothesis as

$$\tilde{\tau}_{ij} = \mu \left(\frac{\partial \tilde{u}_i}{\partial x_j} + \frac{\partial \tilde{u}_j}{\partial x_i} \right) - \frac{2}{3} \mu \frac{\partial \tilde{u}_k}{\partial x_k} \delta_{ij} \quad (3)$$

$$\begin{aligned} \frac{\partial \tilde{E}}{\partial t} &= - \frac{\partial}{\partial x_i} (\tilde{E} \tilde{u}_i + p \tilde{u}_i) \\ &+ \frac{\partial}{\partial x_i} \kappa \frac{\partial \tilde{T}}{\partial x_i} - \frac{\partial (C_p q_i)}{\partial x_i} + \frac{\partial}{\partial x_i} (t_{ij} + \tau_{ij}) \end{aligned}$$

$$\begin{aligned} &+ \frac{\partial}{\partial x_i} \left(\bar{\rho} \sum_{j=1}^{NumSpecies} \bar{H}_k (D_k + \frac{\mu_{sgs}}{Sc_t}) \frac{\partial \tilde{Y}_k}{\partial x_i} \right) \\ &+ \bar{\rho} \sum_{j=1}^{NumSpecies} \tilde{Y}_j f_j(x_i) \cdot \tilde{u}_i - \sum_{k=1}^{NumSpecies} H_{o,k} w_k \end{aligned} \quad (4)$$

$$\begin{aligned} \frac{\partial \bar{\rho} \tilde{Y}_j}{\partial t} &= - \frac{\partial}{\partial x_i} (\bar{\rho} \tilde{u}_i \tilde{Y}_j) + \frac{\partial}{\partial x_i} \bar{\rho} \left[\left(D_{j,N} + \frac{\mu_{sgs}}{Sc_t} \right) \frac{\partial \tilde{Y}_j}{\partial x_i} \right] \\ &+ \bar{\rho} \tilde{Y}_j f_j(x_i) + w_{chem}^j + w_{spray}^j \end{aligned} \quad (5)$$

The SGS stress tensor τ_{ij} and SGS heat flux vector q_i in Eqs. (2) and (4) are defined respectively as

$$\tau_{ij} - \frac{1}{3} \tau_{kk} \delta_{ij} = -2\mu_{sgs} \left(\tilde{S}_{ij} - \frac{1}{3} \tilde{S}_{kk} \delta_{ij} \right) \quad (6)$$

$$q_j = - \frac{\mu_{sgs}}{Pr_{sgs}} \frac{\partial \tilde{T}}{\partial x_j} \quad (7)$$

where μ_{sgs} is the SGS viscosity, Pr_{sgs} is the SGS Prandtl number, and $\tilde{S}_{ij} = \frac{1}{2} \left(\frac{\partial \tilde{u}_i}{\partial x_j} + \frac{\partial \tilde{u}_j}{\partial x_i} \right)$ is the strain rate tensor. Here $\tilde{\cdot}$ is a Favre-filtered variable obtained from its grid-filtered component $\bar{\cdot}$. In this study, the box or top hat filter is applied for the grid-filtered component.

Dynamic Vreman SGS LES model

The development of the dynamic subgrid-scale model (DSGS) model reflects significant progress in the subgrid-scale modeling of non-equilibrium flows. The DSGS model calculates the model coefficient from the energy of the smallest resolved scale, rather than by setting a priori parameters. The DSGS is obtained by two filtering processes: in the first one, the grid filter Δ is applied, the filtered expressions are given by (1)-(4), where the SGS Reynolds stress was included. Then adding a test filter $\hat{\Delta} = 2\Delta$ to the grid filtered equations (1)-(4) leads to the subtest-scale stress tensor T_{ij} and subtest-scale heat flux vector Q_j :

$$T_{ij} - \frac{1}{3} T_{kk} \delta_{ij} = -2\mu_{sgs} \left(\hat{S}_{ij} - \frac{1}{3} \hat{S}_{kk} \delta_{ij} \right) \quad (8)$$

and

$$Q_j = - \frac{\mu_{sgs}}{Pr_{sgs}} \frac{\partial \hat{T}}{\partial x_j}, \quad (9)$$

here we define $\mu_{sgs} = \bar{\rho} C_{DVMG} \Pi^t$ and $Pr_{sgs} = Pr_{DVMG}$. With Germano identity [14] and the least-squares error minimization technique of Lilly [15], the coefficients C_{DVMG} and Pr_{DVMG} are obtained as

$$C_{DVMG} = \frac{\langle L_{ij} M_{ij} \rangle_V}{\langle M_{ij} M_{ij} \rangle_V} \quad (10)$$

and

$$Pr_{DVMG} = \frac{\langle M_j^\theta M_j^\theta \rangle_V}{\langle L_j^\theta M_j^\theta \rangle_V}, \quad (11)$$

PARALLEL TECHNIQUE

Parallel Solution of the Semi-implicit Scheme for Mass, Momentum and Energy

For the FEM discretization on the decomposed domain, the elements dissected are considered overlapped. Some information used for the material properties and primitive variables integration on an overlapping element must be gathered. This is a message passing gather in MPI. In FEM, for each node, we need the integration over elements, on which this node resides. The integration over an overlapped element requires gathering values whenever a node is off processor. Therefore, we need to know for each processor,

1. which elements are overlapped, this is Elem_Shared 1d array with the dimension of shared elements.
2. how many nodes should be received from each other processors and this is COMMONR 1d array with the dimension of number of processors.
3. how many nodes should be sent to each other processors and this is COMMONS 1d array with the dimension of number of processors.
4. which nodes should be sent to which processor, and this is the global_sharednode 2d array.
5. what is the new order for nodes and elements in terms of the global numbering, which are represented by NEWORDER and NEWORDER_E.

In the explicit case, we merely update the new time value at n+1 and no matrix equation needs to be solved. Elements that are needed to be integrated are elements residing on this processor and shared elements from Elem_Shared, and only update nodes on this processor according to the array NEWORDER, when it is not zero and the value is less than the number of nodes on that processor, denoted by nnode. Denote nele as the number of elements residing on this processor and nElem_Shared as the number of shared elements on that processor, then All_nElem=nele+nElem_Shared which is how many elements will need to be integrated for each processor. Then we need a new connectivity NODE, of which the row is All_nElem. The column of NODE will include local node and shared node from Node_Shared. This process is described in following Fortran pseudo code:

```
DO K = 1, All_nElem
  DO KK=1,number of nodes for each element
```

```
    L=NODE(K, KK)
    IF (L.NE. 0 .AND. L.LE. nnode) THEN
      Update node L
    END IF
  END DO
END DO
```

After all nodes on that processor are updated, we need to pass shared nodes to their corresponding processors which can be found in array global_sharednode and be done by MPI_gatherv or MPI_Allgatherv command. In our code, MPI_Allgatherv is used so that every processor gets the same gathered value from all processors and when the processor is the receiving processor, only the shared nodes gets updated. Here sharednnode is used to track the amount of shared nodes. This step has to be done before going into next time step. This is applicable to the momentum predictor, and the corrector, along with all transport equations. Fortran pseudo code is given as:

```
DO I=1, number of total processors
  sBuf(:, :)=0
  DO J=1, COMMONS(I)
    II=NEWORDER(global_sharednode(I, J))
    sBuf(I, J)=Var(II)
  END DO
  rBuf(:)=0
  call MPI_Allgatherv to gather
  sBuf(I, 1:COMMONS(I)) to rBuf(:) onto
  all processors
  IF (rank of this processor==I-1) Then
    DO J=1, sharednnode
      Var(nnode+J)=rBuf(J)
    END DO
  END IF
END DO
```

Where Var is for any variable that is needed to be updated, i.e. velocity U or density Rho.

For the implicit solution processes for pressure the PCG linear equation solver package having various Krylov solvers with user supplied matrix-vector multiplication (matvec) and dot product operations is incorporated. In addition, the PCG package provides for user developed equation preconditioning and overall equation convergence. In this case we use a Beam-Warming method for evoking the solution to the equations that uses an additive Schwartz preconditioning system. Only the communication for preconditioning requires the matrix-vector multiple (matvec) and vector dot product be user supplied, and these require a collocation, that is GATHERV and ALLREDUCE processes from MPI with the help of array global_shared node. The Beam-Warming iteration process provides for the global convergence of the equation system as developed and described by Carrington [16].

Spray model

In the computational domain there are two separate phases present, namely the continuous (fluid) and the discrete phase (particles). Equations (1)-(5) are solved for the continuous phase only and the motion of particles is dealt with particle trajectory calculations. Through an iterative solution procedure the mass, momentum and energy interaction between both phases can be realized. In order to simulate spray formation, (discrete) liquid particles have to be introduced to interact with the present (continuous) gas phase. In diesel sprays the primary breakup takes place in the atomization regime. So, it is assumed in our paper that there is no liquid core; all the liquid is formed into droplets immediately after the exit of the nozzle hole. That is where the so-called atomizer model comes into play. The atomizer creates initial conditions, that depend on the internal nozzle flow, for further particle trajectory calculations by defining initial droplet diameter, velocity and the cone angle of the spray.

We think of sprays as being a cloud of diffuse particles; 10's of thousands of droplets of various sizes streaming through what is usually a gaseous media. These particle clouds proceed through numerous processes of agglomeration and break-up as they move through the background or conveying fluid (often air). The droplets experience interactions with this conveying media where stresses on the droplets force breakup and agglomeration. While these stresses are composed of many components, drag is most important, with turbulent changes in these forces too, needing accounting as does evaporation. The conveying media also experiences forces governed by Newton's 3rd law, working to move the fluid and dampen its turbulence.

We formulate a Boltzmann's type transport equation describing droplet transport. Starting with the probable number of droplets in a volume at any time, t , is mathematically stated as

$$P = f(x_i, u_i, r, q) dvdr \quad (12)$$

The vector and scalar values represent the droplets spatial coordinates x_i , velocity components u_i , radius r , and source terms q . This model equation applies for very small droplets and assumes they have a spherical shape -- reasonable for very small drops. Most sprays are composed of many sizes of droplets, particularly true late in the transport process, or the secondary breakup phase. The primary breakup phase is the portion of the spray's development being addressed by the proposed research, and will give rise to accurate phase-space information needed to solve the spray equation. Droplets are continuously interacting so, the probable number of droplets changes in time and space. By taking the total derivative of P a spray equation introduced by Williams [17] is

$$\frac{\partial f}{\partial t} = -\frac{\partial}{\partial x_i}(u_i f) - \frac{\partial}{\partial u_i}(F_i f) - \frac{\partial}{\partial r}(Rf) + Q_f \quad (13)$$

This describes the change in the probable number of droplets in space and time as a function of velocity, accelerations, change in size, with included sources. In our work, we solve this equation

with a combination of engineering models while providing advective transport with a stochastic Lagrangian Particle Transport (LPT) method [10,11]. This has been shown to work with reasonable accuracy and efficacy once adjusted for specific injectors. An overview of the solution process and models follows, allowing us to expose what is missing from the solution process and required from the multiscale breakup process, known as the initial breakup phase.

Droplet sizes changes occur when they breakup, agglomerate and evaporate. A nondimensional model for convective rate of heat and mass transfer is employed. Conduction heat transfer in the droplets is incorporated to determine the interior temperature distribution. Mostly drag forces are present on the particles, and these are estimated with standard drag coefficients. Newton's 3rd Law allows for the evaluation of opposing forces acting on the conveying fluid. In the vicinity of a tiny droplet, isotropic turbulence can be assumed and is well represented by a Gaussian distribution. Turbulent dispersion is employed when the particle's transverse velocity is larger than the local eddy break-up time. If particles transverse slowly, less than the eddy break-up time, the particles are modeled as trapped by turbulent eddies. Models for agglomeration and break-up along with completely elastic collisions are functions of surface tension, liquid viscosity, impact momentum and geometric shape. More details about break-up and collisions models can be found in [9]. Therefore, for most sprays we introduce the model of droplet distortion rate, \tilde{y} and the rate of change of \tilde{y} , given as $\tilde{\dot{y}}$. With these considerations, the droplet rate function in KIVA [9-12] becomes

$$\begin{aligned} \frac{\partial f}{\partial t} = & -\frac{\partial}{\partial x_i}(u_i f) - \frac{\partial}{\partial u_i}(f F_i) - \frac{\partial}{\partial r}(Rf) \\ & - \frac{\partial}{\partial T}(\tilde{T}f) + \frac{\partial}{\partial y}(\tilde{y}f) - \frac{\partial}{\partial \tilde{y}}(\tilde{\dot{y}}f) \\ & + \tilde{f}_{break} + \tilde{f}_{collide} \end{aligned} \quad (14)$$

Shown now are models for break-up \tilde{f}_{break} and collision $\tilde{f}_{collide}$ (that includes elastic collisions and agglomeration), rate of change in radius (evaporation), rate of oscillation \tilde{y} (oscillation velocity), time rate of oscillation velocity (oscillation acceleration) $\tilde{\dot{y}}$, and rate of change of droplet temperature, \tilde{T} . The temperature is included in the transport solution because viscosity, surface tension, and partial pressures exerted by the liquid droplet all vary as a function of droplet temperature. Mechanisms of advection and turbulent influences are employed in the first two terms, on the right hand side of Eq. (14). These two terms are combined into the Lagrangian particle technique, a stochastic method with a deterministic advection component. Conservation of mass, momentum and energy is maintained with the gaseous

fluid, forming source terms for the conveying gaseous fluid in mass, species, momentum, and energy. Solving the spray equation produces these conservation values.

Parallel Solution for the spray model

In our work, we associate all parcels to their element location, which is defined by the parcels position. Therefore parcels will be distributed across different processors according to the decomposition of elements. Here we include local elements and shared elements on one processor. Calculation of the parcels properties will only be done on the processors that have those parcels. One processor (I/O processor) will read in the new parcels and distribute all parcels to their corresponding processors according its location element. Thus, every processor has its own amount of parcels, denoted as numpart. When one parcel moves to another element which resides on a different processor, all properties of that parcel will be passed to the new processor from its previous processor and this parcel will be removed from its previous processor and added to the new processor. Therefore, every time step, we need to reorder parcels on each processor: subtract the ones that are neither in local element nor in the shared elements. This process can be illustrated in the following steps:

1. In I/O processor, we read in all new parcels and its location elements (global element) will be recorded in an array named newINJ_PART_ELEM
2. Distribute new parcels to their corresponding processors. For each processor, if NEWORDER_E(newINJ_PART_ELEM(I)) is not equal to zero, then this new parcel belong to this processor and numpart=numpart+1.
3. For each processor, one dimension arrays lmovepar and gmovepar will be used to track which parcel has been moved. lmovepar is the local order of that parcel while gmovepar is the global order. The amount of moved parcels is kept by nmovepar. Which element this parcel moved to will be stored in a 1d array toelement, which is global value.
4. Using those arrays from step 3, for each processor, we can find out which local parcel is sent to which processor, and this is the S_Part 2d array. It keeps the local parcel's global order.
5. For each processor, how many parcels will be received from which processor will be stored in a 1d array R_npart, and how many parcels will be sent to which processor is in a 1d array S_Npart.
6. Update all new parcels for each processor. This can be done with this Fortran pseudo code:

```
DO I=1, number of total processors
  sBuf(:, :)=0
  DO J=1, S_Npart(I)
```

```
    II=S_Part(I,J)
    sBuf(I,J)=Var(II)
  END DO
  rBuf(:)=0
  call MPI_Allgatherv to gather
  sBuf(I,1:S_Npart(I)) to
  rBuf(:) onto all processors
  IF(rank of this processor==I-1)Then
    DO J=1, R_npart(I)
      NewVar(J)=rBuf(J)
    END DO
  END IF
END DO
```

Where Var is for any property of parcels that is needed to be updated, e.g the location or velocity of the parcel, and NewVar is the updated value for new parcels in that processor

7. After receiving or sending parcels between processors, each processor need to reorder its parcels. For each processor, which parcel has been moved out, this is nRemoveP and what they are, this is 1d array RemoveP. The Fortran pseudo code is:

```
DO I=1, nmovepar
  IF(NEWORDER_E(toelement(I))==0) THEN
    nRemoveP=nRemoveP+1
    RemoveP(nRemoveP)=lmovepar(I)
  END IF
END DO
```

After putting RemoveP in descending order, we are ready to reorder all parcels on each processor.

NUMERICAL RESULTS

3-D flow over a cylinder by LES

3-D Turbulent flow over a circular cylinder for Reynolds numbers 1000 is investigated here using our proposed dynamic LES FEM. The vortex shedding is observed. Here we used 687,616 nodes and 642,288 elements. The domain is decomposed onto 36 subdomains (processors). The domain decomposition is done by ParMETIS, which is an MPI-based parallel library that implements a variety of algorithms for partitioning unstructured graphs, meshes, and for computing fill-reducing orderings of sparse matrices. The domain decomposition is shown in Fig. 1 (a) and the simulation mesh set up is as Fig. 1 (b). LES requires finer mesh in order to approximate the boundary layer better. Therefore the mesh is finer around the cylinder. The simulation setup is: The dimension of the domain is $x \in [-0.275m, 0.275m]$, $y \in [-0.1375m, 0.1375m]$ and $z \in [-0.165m, 0.165m]$. Inflow $U=1.0m/s$, the cylinder diameter is 0.02m, free outflow and no

slip boundary condition on the wall and periodic boundary condition on the Z direction wall so that the 3d effects from the side walls can be mitigated. Fig. 2 shows the velocity contour slice at $Z=0$ for $Re = 1000$ at 0.7s and 1s to demonstrate the vortex shedding. From Fig. 2, we can see the flow starts to separate from $\theta = 80^\circ$ which agrees with other literatures [18]. Fig. 3 shows the 3D velocity streamline around the cylinder at 0.7s and 1s. Our parallel dynamic Vreman LES model can capture vortex shedding without any wall functions and ad-hoc adjustment and on a LANL Wolf Intel E5-2670, the solving time for each time step can be reduced to 2 seconds, among which 1 second is used for solving the pressure implicitly. A scaling test has been done with a smaller set of elements: 150K nodes onto 1 to 32 processors. We can see a maximum speed-up of 21 times faster than serial in Fig. 4; refer to Waters and Carrington [19] for more details about scaling. The results shown in Fig. 4 are plotted on a log-log scale and we show essentially linear or slightly superlinear scaling; attributable to how the domains share and pass information in our scheme, being highly vectorized. In order to minimize the communication cost, an optimal number of nodes on each processor should be met.

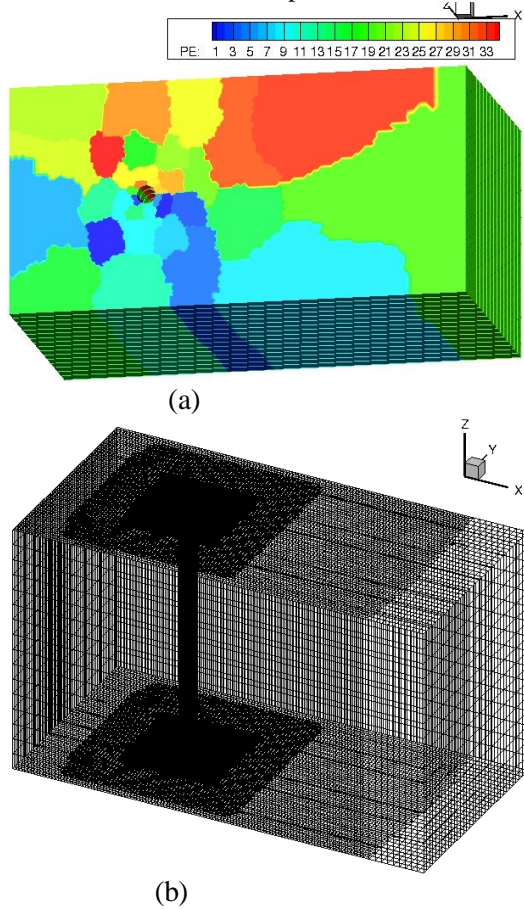


Fig. 1: (a) Simulation domain decomposition onto 36 processors and (b) mesh set up

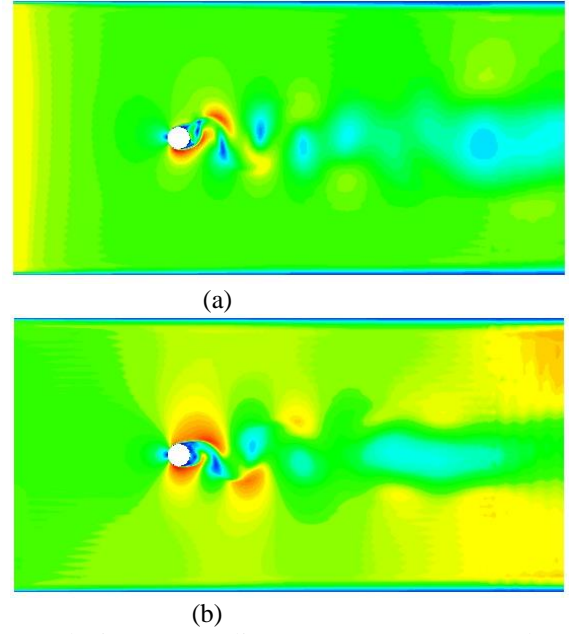


Fig. 2: Velocity contour slice at $Z=0$ at (a) 0.7 seconds and (b) 1.0 second.

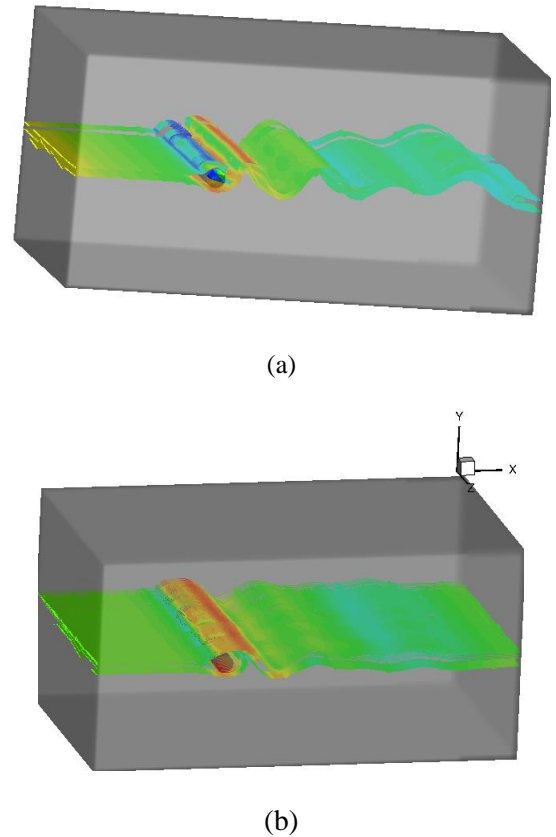


Fig. 3: 3D velocity streamline at (a) 0.7 seconds and (b) 1.0 seconds.

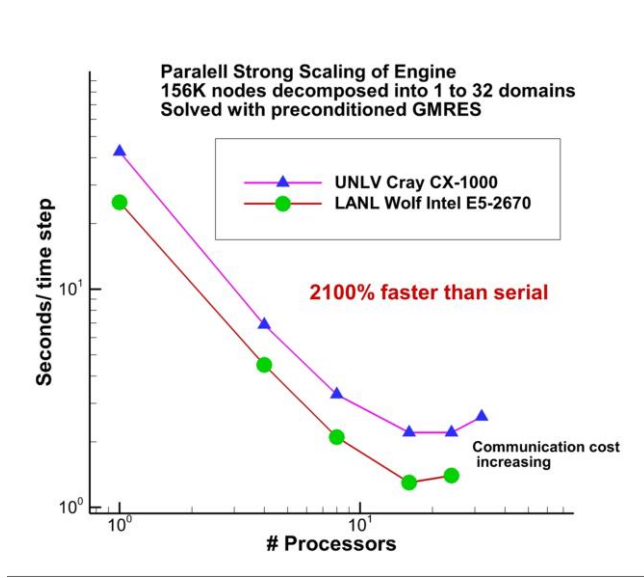


Fig. 4: Parallel Strong Scaling

The second flow over cylinder test is done for Reynolds numbers 1.2×10^5 with the same mesh set up and domain decomposition as in Fig. 1. In order to compare the pressure coefficient with the experimental data from Merrick and Bitsuamlak[20], we have the same set up as the experiments shown in [20]. We compared the pressure coefficient by our method with the experimental data from Merrick and Bitsuamlak [20] in Fig. 5 and they reach good agreement. It also shows the boundary layer detachment occurs at $\theta = 80^\circ$, which is true for subcritical Reynolds number [18]. Fig.5 demonstrates the continuous development of the wake's streamlines as the time evolves. The wake started in a symmetric fashion Fig.6 (a), then the secondary eddies forms at the downstream of the point of boundary layer separation and boundary layer started to detach from the cylinder surface, creating the onset of the separation process Fig. 6 (b). Therefore the symmetry got broken and eddies are shed from the cylinder and the familiar steady flow pattern of periodic vortex shedding is initiated. At later time, the vortices of the wake generate large recirculation in the rear of the cylinder and develop into the oscillatory Von-Karman wake Fig.6 (c)-(f). Those simulation results agree with Mustto and Bodstein [21].

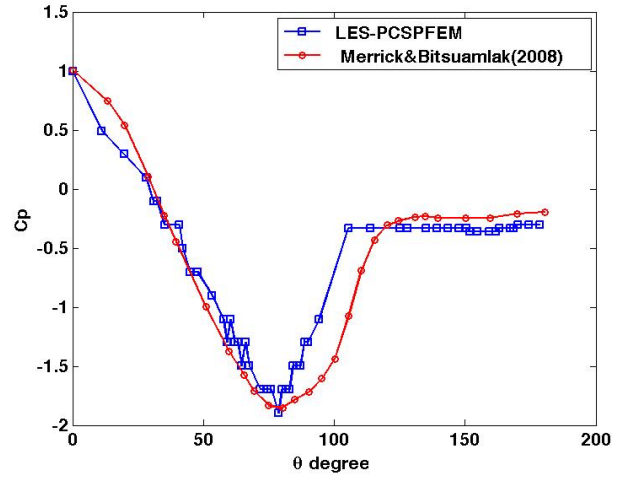


Fig 5: Pressure Coefficient for $Re= 1.2 \times 10^5$

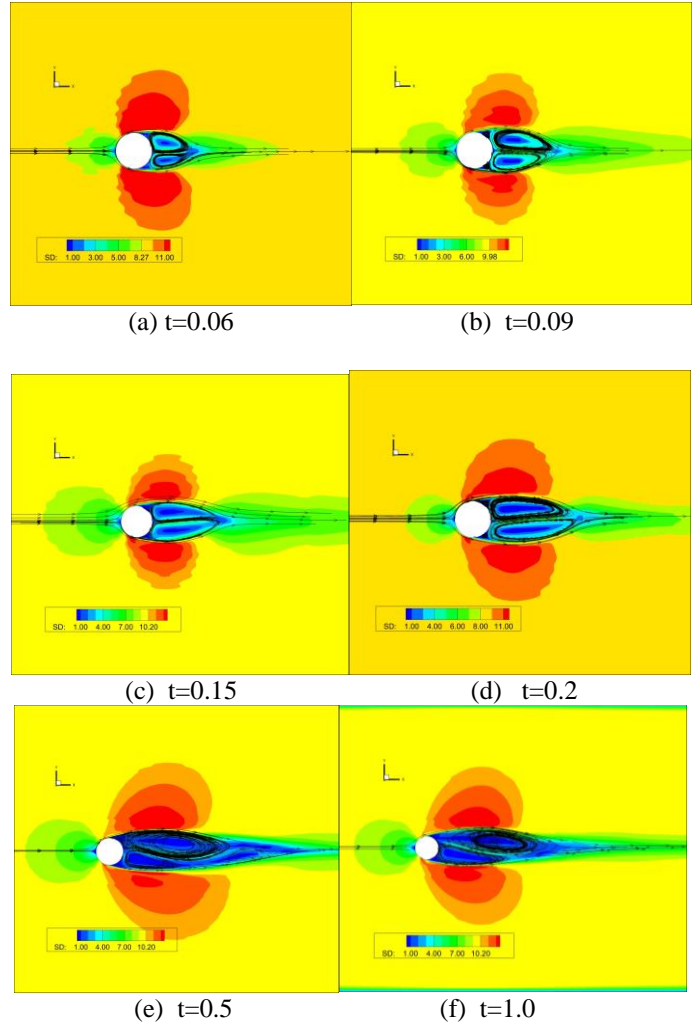


Fig. 6 Instantaneous velocity streamlines at different time (seconds) of the simulation; $Re=1.2 \times 10^5$

Spray Modeling Verification and Validation

In order to validate our parallel spray model, Diesel was injected into different quiescent Nitrogen pressure of 1, 11, 30 and 50 atmos. The dimension of the domain is $x \in [-0.022m, 0.022m]$, $y \in [-0.022m, 0.022m]$, $z \in [0, 0.55m]$. Velocity of injected spray ranges 85 m/s and 115m/s. The results of spray penetration versus time from our parallel LES method are compared with Experimental data from Hiroyasu and Kadota [22] and KIVA-hpFE serial code and adapted grid by Carrington et. al. [23, 24]. We obtained good agreement with experimental data and other methods by KIVA in Fig. 7. With our Finite Element Spray modeling, we can get 2nd order accuracy even on coarser grids. Since our grid is coarse, the domain was decomposed onto only 3 processors in order to test the accuracy of our parallel mechanism for particles Fig. 7. The injector is located at the top of the domain which is in processor 2. Fig. 8 is showing the domain decomposition on 3 different processors. The velocity of injected spray is shown in Fig. 9 at different pressure 1.1 MPa and 5.0 MPa.

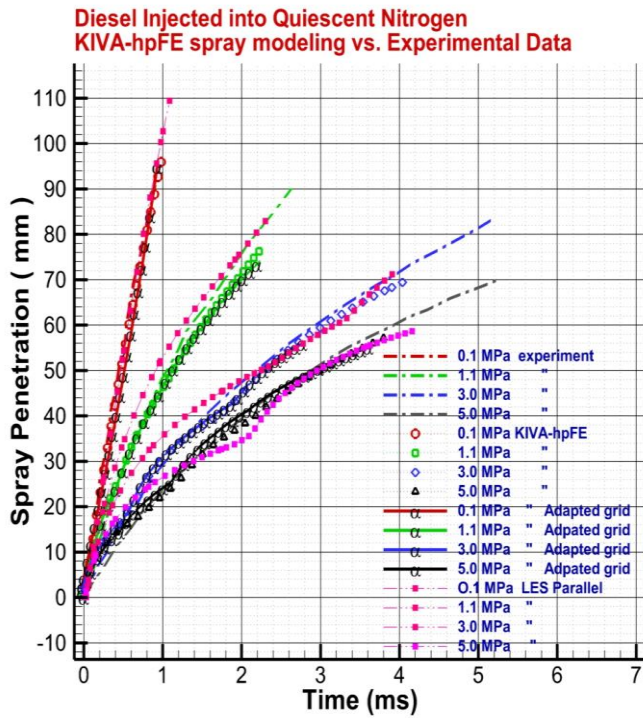


Fig. 7 Spray penetration versus time

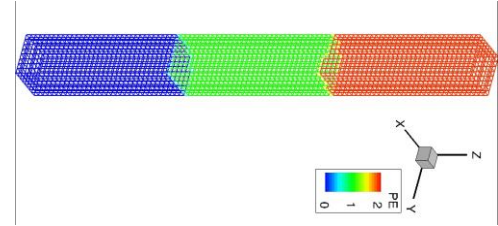
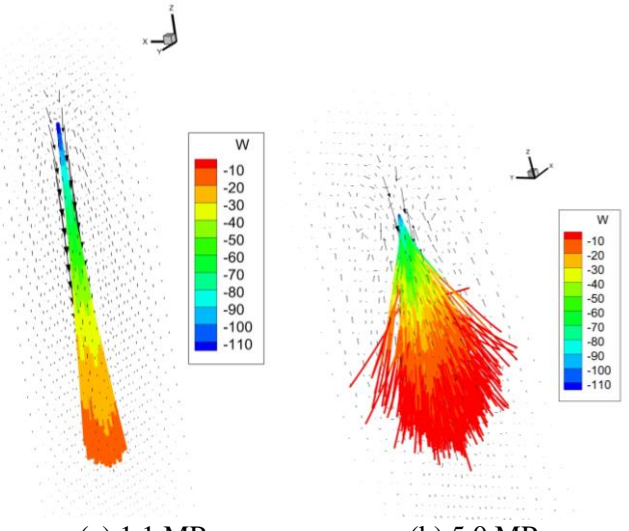


Fig. 8 mesh decomposition



(a) 1.1 MPa (b) 5.0 MPa
Fig. 9 Velocity of injected spray at pressure
(a) 1.1 MPa and (b) 5.0 MPa

DEVELOPMENT OF ENGINE SIMULATION

Simulating engines is performed with moving parts that are represented by overset surface grids (triangular elements which markers and normal vectors). The specifics on local-ALE method is detailed in [25-27] where the errors are calculated and described showing the system to be 2nd order accurate in space for momentum and scalar transport. Grid generation is simplified considerably with the local-ALE scheme and use of the overset grid; the moving parts surfaces simply are overlaying the hexahedral mesh produced by most grid generators, most of which can do a reasonably quick and good hex grid on convex domains without cutouts for valves, piston geometries.

The moving parts system is shown in Fig. 10 with the cylinder grid and overset grid parts for a scalloped bowl piston and 2 valves in exploded view. The moving parts are triangulated surfaces. In Fig. 11 a and b you'll notice the displacement of the piston (now overlaying the cylinder grid) and motion of the fluid. The grid generator for hexahedral elements only need concern for the the full cylinder and port convex cavities; there is no need for cutouts and working around the moving parts.

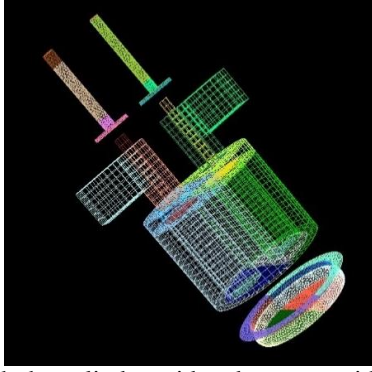


Fig. 10 with the cylinder grid and overset grid of moving parts in exploded view.

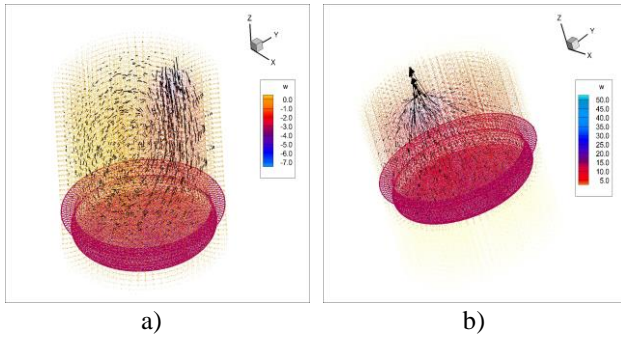


Fig. 11 Scalloped blow piston moving in pump/engine
a) intake stroke and b) exhaust stroke

Combined with reactive chemistry from KIVA's legacy system, we are approaching the ability to model complete internal combustion engines robust and accurate. The LES and RANS fully coupled implicit momentum, energy and turbulence modeling has been achieved. Results combustion of a rich mixture of gasoline of a carbureted engine is shown in Fig 12. The simulation is using the $k-\omega$ two-equations RANS mode where the concentration of O_2 is shown with burn at TDC. The mixture was ignited by simulation a newly developed plasma-kernel spark model applied at a point (very small volume) in the domain. The simulation results here are illustrative showing the capability, and presumably with all the validation we've made on each portion of the system, the entire system is properly functioning, and it appears so but further analysis and comparison to experimental data of some benchmark engines is still to be completed.

CONCLUSION

This paper provides a parallel mechanism for the Finite Element Method based algorithm solving 3D turbulent reactive flow in engines. Dynamic LES are investigated on 3d Flow over a cylinder for Reynolds number 10^3 and 10^5 with our parallel (MPI) algorithm. Other published research was also presented to support the validity of the results. Given a fine mesh for the 3D simulation, our parallel algorithm on FEM is able to perform the calculation in an enhanced calculation speed, 2 seconds per time

step for a mesh with 687,616 nodes and 642,288 elements on 36 processors. The efficiency of the parallel algorithm has been proved in our previous work. A detailed explanation of the parallel scheme for particles is also given in this paper and it is tested to validate our spray model. The results are compared against experimental data and they are in good agreement. This paper sets up a good foundation for solving turbulent reactive flow in engines with a spray model.

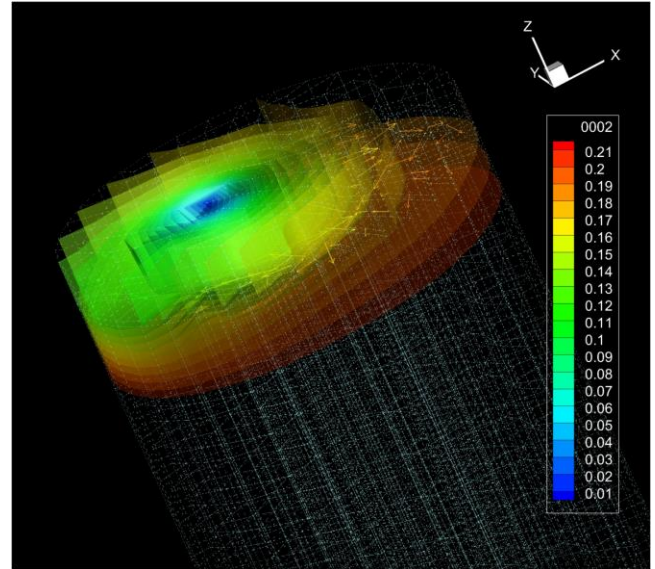


Fig. 12 Combustion of gasoline in an engine at TDC, with isopleths showing concentration of O_2 a few microseconds after burn initiation.

ACKNOWLEDGMENTS

The DOE's Office of Energy Efficiency and Renewable Energy (EERE) Advanced Combustion Program (Gurpreet Singh and Leo Breton) is supporting this effort. Los Alamos National Laboratory, an affirmative action/equal opportunity employer, is operated by the Los Alamos National Security, LLC for the National Nuclear Security Administration of the U.S. Department of Energy (DOE) under contract DE-AC52-06NA25396. Los Alamos National Laboratory strongly supports academic freedom and a researcher's right to publish; as an institution, however, the Laboratory does not endorse the viewpoint of a publication or guarantee its technical correctness.

REFERENCES

- 1) Desjardins, P. E. and Frankel, S. H., 1999, "Two dimensional Large Eddy Simulation of soot formation in the near field of a strongly radiating non-premixed acetylene-air jet flame", *Combust. Flame*, 119, 121-133.
- 2) Colin, O., Ducros, F., Veynante, D. and Poinso, T., 2000, "A thickened flame model for large eddy simulations of turbulent premixed combustion", *Phys. Fluids*, 12, 1843-

1863.

- 3) Angelberger, C., Egolfopoulos, F. and Veynante, D., 2000, "Large Eddy Simulations of chemical and acoustic effects on combustion instabilities", *Flow Turb. and Combustion*, 65, 205-22.
- 4) Pitsch, H. and Duchamp, de la Geneste L., 2002, "Large Eddy Simulation of Premixed Turbulent Combustion using a level-set approach", *Proceedings of the Combustion Institute*, 29, in press.
- 5) Pierce, C. D. and Moin, P., 2004, "Progress-variable approach for large eddy simulation of non-premixed turbulent combustion", *J. Fluid Mech.*, 504, 73-97.
- 6) Lau, G. E., Yeoh, G. H., Timchenko, V. and Reizes, J. A., 2012, "Application of dynamic global-coefficient subgrid-scale models to turbulent natural convection in an enclosed tall cavity" *Physics of Fluids* (1994-present), 24, 094105.
- 7) Vreman, A. W., 2004, "An eddy-viscosity subgrid-scale model for turbulent shear flow: Algebraic theory and applications," *Physics of Fluids*, 16, pp. 3670-8.
- 8) Waters, J., Carrington, D.B. and Pepper, D.W., An Adaptive Finite Element Method with Dynamic LES for Turbulent Reactive Flows," *Computational Thermal Sciences*, to appear in 2016.
- 9) Amsden, A. A., O'Rourke, P. J., Butler, T. D., 1989, "KIVA-II, a computer program for chemically reactive flows with sprays," Los Alamos, N.M. : Los Alamos National Laboratory Scientific Report, LA-11560-MS.
- 10) Dukowicz, J.K, April 1980, "A Particle-Fluid Numerical Model for Liquid Sprays," *Journal of Computational Physics*, vol. 35, 2.
- 11) Torres, D.J., O'Rourke, P.J., Amsden, A.A., , 1 2003, "Efficient multicomponent fuel algorithm," *Combustion Theory and Modelling*, vol.7, p.66-86
- 12) Torres, D.J. and Trujillo, M.F., 2006, "KIVA-4: An unstructured ALE code for compressible gas flow and sprays," *Journal of Computational Physics*, Elsevier, pp. 943-975.
- 13) Jimack, P. K., and N. Touheed, 2000, "Developing parallel finite element software using MPI", *HPC Comp. Mech*:15-38.
- 14) Germano, M., Piomelli, U., Moin, P., and Cabotm , W. H., 1991, "A dynamic subgrid-scale eddy viscosity model," *Physics of Fluids A*, vol. 3, no. 7, pp. 1760-1765.
- 15) D. K. Lilly, 1992, "A proposed modification of the Germano subgrid-scale closure method", *Physics of Fluids A*, vol. 4, no. 3, pp. 633-635.
- 16) Carrington, D.B., January 2008, "A Parallel First-Order Spherical Harmonics (P_1) Matrix-Free Method for Radiative Transport," *Numerical Heat Transfer, Part B: Fundamentals*, Vol. 53, pp. 1-21, Taylor and Francis.
- 17) Williams, F.A., "Spray Combustion and Atomization," *Physics of Fluids*, vol 1, 6, 1958, p. 541-545.
- 18) Kawamura, T., Nakao, T., Takahashi, M., Hayashi, M., Murayama, K., Gotoh, N., 2003, "Synchronized Vibrations of a Circular Cylinder in Cross Flow at Supercritical Reynolds Numbers", *ASME. J. Pressure Vessel Technol.* 125(1):97-108.
- 19) Waters, J. and Carrington, D.B., "A Parallel Large Eddy Simulation in a Finite Element Projection Method for All Flow Regimes", *Numerical Heat Transfer, Part A: Applications*, accepted in 2016.
- 20) Merrick, R., and Bitsuamlak, G., 2008, "Control of flow around a circular cylinder by the use of surface roughness: A computational and experimental approach." *Internet publication at http://www.ihrc.fiu.edu/wpcontent/uploads/2014/03/MerrickandBitsuamlak_FlowAroundCircularCylinders.pdf*.
- 21) Mustto, A. A. and Bodstein, G. C. R., 2011, "Subgrid-Scale Modeling of Turbulent Flow Around Circular Cylinder by Mesh-Free Vortex Method", *Engineering Applications of Computational Fluid Mechanics*, 5:2, 259-275, DOI: 10.1080/19942060.2011.11015369.
- 22) Hiroyasu, H. and Kadota, T., 1977 "Fuel Droplet Size Distribution in Diesel Combustion Chamber," SAE paper 740715.
- 23) Carrington, D.B., Wang, X. and Pepper, D.W., 2010, "An h-Adaptive Finite Element Method for Turbulent Heat Transfer," *Computer Modeling in Engineering & Sciences*, vol 61, no 1, pp. 23-44. Tech Science Press.
- 24) Carrington, D.B., Wang, X. and Pepper, D.W., 2013, "A Predictor-Corrector Split projection method for turbulent reactive flow", *Comput. Therm. Sci.*, vol. 5, 4, pp. 333-353.
- 25) Carrington, D.B., Munoz, D., Heinrich J.C., "A local ALE for flow calculations in physical domains containing moving interfaces," *Progress Computational Fluid Dynamics, an Int. Jour.* vol 14, no, 3, pp. 139-150, 2014
- 26) Hatamipour, V.D., Carrington, D.B., Heinrich, J.C., "Accuracy and Convergence of Arbitrary Lagrangian-Eulerian Finite Element Simulations based on a Fixed Mesh," *Progress in Computational Fluid Dynamics, an Int. Jour* (submitted December 2015).
- 27) Carrington, D.B., Mazumder, M., Heinrich, J.C., "Three-Dimensional Local ALE-FEM Method for Fluid Flow in Domains Containing Moving Boundaries/Objects Interfaces," *Progress in Computational Fluid Dynamics, an Int. Jour.* (submitted December 2015)

Three-dimensional ALE-FEM method for fluid flow in domains with moving boundaries part 1: algorithm description

David B. Carrington*

Los Alamos National Laboratory,
Theoretical Division T3,
Los Alamos, NM 87545, USA
Email: dcarring@lanl.gov
*Corresponding author

A.K.M. Monayem Hossain Mazumder

Department of Mechanical Engineering,
Lamar University,
Beaumont, TX 77710, USA
Email: monayem59@gmail.com

Juan C. Heinrich

Department of Mechanical Engineering,
University of New Mexico,
Albuquerque, NM 87131, USA
Email: heinrich@unm.edu

Abstract: A three-dimensional finite element method for simulating fluid flow in domains containing moving objects or boundaries is developed. This method is a type of arbitrary-Lagrangian-Eulerian, based on a fixed mesh that is locally fitted at the moving interfaces and recovers its original shape once the moving interfaces go past the elements. The moving interfaces are defined by marker points so that the global mesh is not affected by the interfaces motion, eliminating potential for mesh entanglement. The result is an efficient and robust formulation for multi-physics simulations. The mesh never becomes unsuitable by continuous deformation, thus eliminating the need for repeated re-meshing. The interface boundaries are exactly imposed Dirichlet type. The total domain volume is always calculated exactly thus automatically satisfying the geometric conservation law. This work supports the internal combustion engines simulator KIVA developed at Los Alamos National Laboratories; in this paper, only the interface moving aspect is addressed.

Keywords: arbitrary-Lagrangian-Eulerian finite element method; time dependent domain; fixed mesh formulation; three-dimensional flow simulations.

Reference to this paper should be made as follows: Carrington, D.B., Mazumder, A.K.M.M.H. and Heinrich, J.C. (2018) 'Three-dimensional ALE-FEM method for fluid flow in domains with moving boundaries part 1: algorithm description', *Progress in Computational Fluid Dynamics*, Vol. 18, No. 4, pp.199–215.

Biographical notes: David B. Carrington is employed by Los Alamos National Laboratory in T-3 Fluid and Solid Mechanics as a Scientist 4, and Team Leader for the development of combustion modelling software. He has worked on numerical methods for heat and mass transfer in fluids and radiation transport at Los Alamos for 15 years. He received his Bachelor, Master and Doctorate in Mechanical Engineering from the University of Nevada, Las Vegas. He is an Associate Editor for the *Journal of Computational Thermal Sciences Thermopedia*.

A.K.M. Monayem Hossain Mazumder received his BS in Mechanical Engineering from Bangladesh University of Engineering and Technology, Dhaka, Bangladesh in 2006, MS in Mechanical Engineering (University of New Orleans, LA) in 2010 and PhD in Mechanical Engineering (University of Oklahoma, Norman, OK) in 2012. As Postdoctoral Fellow at the University of New Mexico, he worked on a method for simulating flow in domains containing moving parts and boundaries. He has worked as a Visiting Assistant Professor at Texas A&M University-Kingsville, Kingsville, TX and currently, working as a Visiting Assistant Professor in Mechanical Engineering at Lamar University, Beaumont, TX.

Juan C. Heinrich received his undergraduate degree from Universidad Católica de Chile and his PhD in Mathematics/Numerical Analysis from the University of Pittsburgh. He is a fellow of the ASME and member of the ASEE, and acts as a consultant to several institutions and companies. He is currently an Editor, Advisor and Reviewer for a variety of technical journals. He began his academic career as a Professor at the University of Arizona, and is currently in the Department of Mechanical Engineering at the University of New Mexico. He has published over 100 technical papers in the area of finite element analysis and is the co-author of two textbooks on the finite element method.

1 Introduction

The accurate numerical simulation of fluid flow in the presence of moving structures that modify the fluid domain is of great practical importance in many areas of engineering and science. These include the automotive, medical, and aeronautics industries. In the automotive industry design refinements through modelling of the combustion inside the engines has become economically important and, in general, internal combustion engines are an important part of our everyday life, being used in a myriad of tools as well as in trains, ships, aircraft and automobiles. An example of modelling software as a tool for the design of internal combustion engines is the Los Alamos National Laboratory engine simulation code 'KIVA' (Torres and Trujillo, 2006). Designs made with these types of software benefit from increased ability to estimate efficiency, power output, environmental impact and other metrics of a prototype before manufacture (Rakopoulos and Mavropoulos, 1996; Yang et al., 2000; Trescher, 2008; Carrington, 2011; Carrington et al., 2014b).

Fluid flow finite element simulators with moving boundaries capabilities are also used in the medical field (Taylor et al., 1998; van Loon and Sherwin, 2006; Kock et al., 2008). The methods used for medical studies generally fall under the category of immersed boundary techniques (Mittal and Iaccarino, 2005; Löhner et al., 2007), these techniques have also been used to investigate the mechanism of insect flight, as small aircraft are pushed to their operational limits at very low Reynolds numbers (Hamamoto et al., 2005; Taira and Colonius, 2009). Finite element numerical models for problems with moving boundaries for the most part are based on arbitrary-Lagrangian-Eulerian (ALE) methods. The earliest ALE applications involved fluid structure interactions, which has been a crucial driver of these efforts especially in aeronautics, also of great interest is a large class of free surface flows (Hirt et al., 1974; Donea et al., 1982; Shyy et al., 1996; Dettmer and Perić, 1996; Farhat and Geuzaine, 2004; Gadala, 2004; Codina et al., 2009; Hua et al., 2011). The ALE process begins at each time step by displacing the boundaries in the Lagrangian framework. The second step is to solve the equations in the fluid domain in the Eulerian framework. The classical ALE approach is to use continuously deforming meshes (Brackbill and Saltzman, 1982; Steger and Benek, 1987; Johnson and Tezduyar, 1994; Askes and Sluys, 2000; Tezduyar, 2001; Saksono et al., 2007). In these schemes, the mesh is attached to the moving boundary, and continuously deformed or

re-generated to adapt to the changing domain geometry during the simulation. There are a variety of other techniques such as embedded mesh, fictitious domain, level set, phase field, etc. that have also been used for many years. However, these do not enjoy the same degree of popularity as ALE methods and will not be addressed here; general discussions on numerical techniques for evolving spatial domains are found in Mittal and Iaccarino (2005), Löhner et al. (2007), Juric and Tryggvasson (1996) and Zhao and Heinrich (2001), where some of the drawbacks of these methods are also discussed.

In finite element method, numerical simulations of fluid flow in the presence of moving solid boundaries, when a moving mesh is used that mesh must be constantly moved and deformed at each time step. The motion and continuous deformation often means that the mesh becomes degraded past the point where it can continued to be used in the calculation, that is, the mesh becomes inadmissible. This is referred to as *mesh entanglement*, and requires the mesh to be periodically regenerated during the simulation. At these times, the program operator must stop and re-mesh, or have a procedure in place that automates this process. Additionally, the variables must be interpolated between meshes, which can lead to instabilities and loss of accuracy and can make the computations expensive (Guillard and Farhat, 2000; Formaggia and Nobile, 2004). This and other practical difficulties make moving mesh schemes undesirable for a variety of practical applications. In this study, the newly developed ALE method eliminates these handicaps by calculating on a mesh that is fixed, the moving interfaces are fitted locally with the geometry deformed only to match the moving interface at its given location; always leaving the global mesh un-deformed. The method addresses the solution of moving boundary problems without utilising adaptive meshes and with no need for periodic re-meshing and interpolation.

The specific purpose of this work is to incorporate the developed numerical capability into the KIVA code for the simulations of internal combustion engines; however, this presentation will be restricted to the description of the mesh moving algorithm, and it will be done in the simpler context of laminar incompressible flow. The algorithm utilises a fixed finite element mesh that at each time step is fitted locally in space and time to conform to the moving interfaces thus preserving the domain volume exactly and therefore is not subjected to the problems associated with the satisfaction of the geometric conservation law associated with moving meshes (Thomas and Lombard, 1979). The

moving interfaces are defined and moved independently using sets of points called ‘marker’ points that are organised in the form of a surface mesh of plane triangular elements that describes the interface; any number of moving interfaces may be present in the domain. A validation of this local ALE method is performed that shows that the algorithm exhibits second order accuracy; a more formal local error analysis has also been performed the details of which are reported in Hatamipour et al. (to be published), the main results of that analysis are mentioned here. Examples showing the flexibility and robustness of the method, as well as its extremely high efficiency are presented, for this purpose the scheme is applied to complex realistic geometries.

In Section 2, the geometric aspects of the interfaces representation and their intersections with the finite element mesh are discussed. Section 3 establishes the equations and finite element formulation used in this work. In Section 4, the mesh fitting algorithm is established. Section 5 presents the results of the method’s validation through a problem with an analytical solution for the flow field, and further applications are presented and discussed in Section 6. This leads to the conclusions in Section 7.

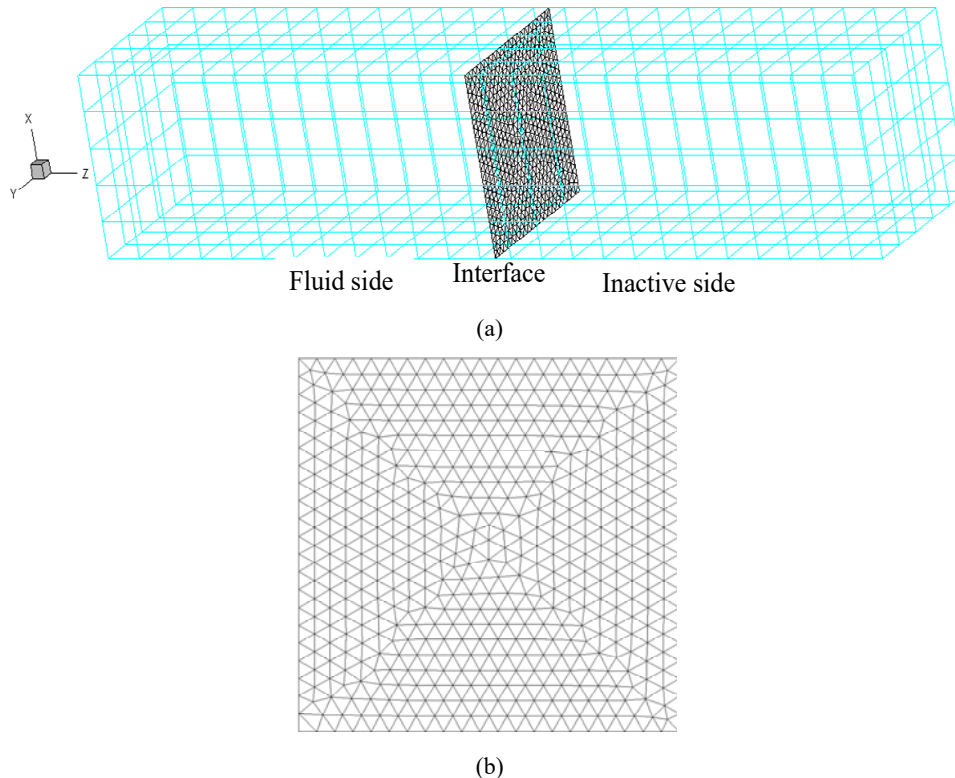
2 Domain and interfaces discretisation

A two-dimensional implementation of the ideas presented here has already been published (Carrington et al., 2014a); the present work concentrates on the three-dimensional

methodology and considers only the case when Dirichlet boundary conditions are imposed on the moving interfaces.

Denote by Ω_T the complete domain occupied by fluid at any time during the full simulation, this domain can also be used as the reference domain in the ALE formulation (Formaggia and Nobile, 2004; Badia and Codina, 2006) and is referred to as the *base* or *reference* domain. Let the domain Ω_T be discretised using a finite element mesh that is deemed as appropriate for all stages of the simulation (the use of adaptive refinement or higher order elements during the simulation is not precluded, but these extensions of the method will not be discussed here). In this work, the mesh is made out of hexahedral tri-linear isoparametric elements; it can also be composed of linear pyramidal elements, which makes the implementation much simpler. The moving interfaces are defined independently using grids of linear triangular surface elements that describe the interfaces in three-dimensional space. The triangular elements that make up the interfaces are called *marker triangles* and their nodes *marker nodes* or *marker points*. The interfaces so defined can move (slide) through the base domain according to their velocity, that is prescribed by a known function or dataset. In this work, it is assumed that the interfaces are rigid and that an interface velocity is prescribed by a given function. Figure 1 illustrates the ideas in a simple hexahedral domain intersected by a plane interface. The interface separates the domain into two parts, one containing the fluid, referred to as the *fluid* portion and the rest of the domain denoted as the *inactive* portion.

Figure 1 (a) Square cylinder discretised by a uniform mesh of 5 by 5 by 25 tri-linear hexahedral elements and intersected by a plane interface perpendicular to the z-axis slightly to the right side of the midpoint in the z-direction (b) Interface intersecting the domain in Figure 1(a); defined by marker triangles with vertices that are marker nodes (see online version for colours)



At each time step in the calculation, an interface intersects the mesh elements at a new position, the intersections of the marker triangles with the element edges are determined and the nodes in the intersected element that are outside the fluid domain are placed on the interface to fit the portion of the element that lies in the fluid, assuming that within an element the intersecting surface is either planar or tri-linear. The interface fitted mesh is used to carry out the flow calculation and once the velocity and pressure are known the adaptation is discarded, the interface position is advanced to the next time step and a new interface fit performed.

Once the moving interface goes past an element, the element regains its original form; the mesh fitting is performed only in those elements intersected by an interface and is local both in space and in time. If Dirichlet conditions are imposed at the interface, the calculation of velocity involves only mesh nodes contained on the fluid side and interpolation is never required; only the pressure needs to be calculated at the interface. Note that the mesh in the inactive part does not enter the calculation and therefore it does not matter how deformed it becomes. This is important in cases involving complex intersections where the mesh in the inactive side may become inadequate for calculation.

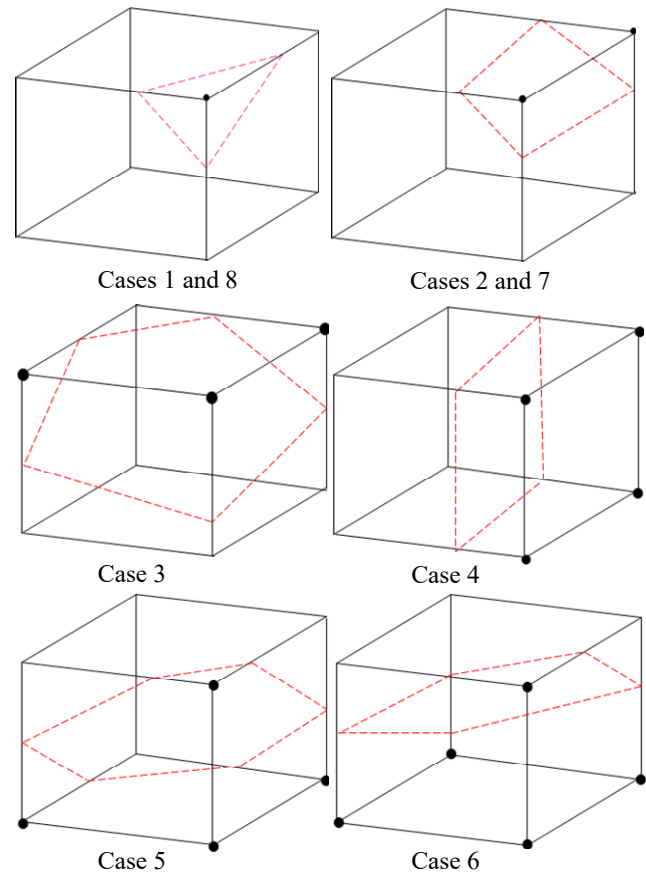
In three-dimensions, there are eight possible different situations that can arise when a hexahedral element is intersected by a plane. These situations are characterised by the number of nodes that are contained in the fluid portion of the element and are listed below:

- Case 1 Pyramidal intersection with one node in the fluid volume.
- Case 2 Prismatic intersection with two nodes in the fluid volume.
- Case 3 Irregular hexahedral intersection with three nodes in the fluid volume, the interface intersects the element at five edges. Two of the element faces in the fluid volume have five corners and two are triangular.
- Case 4 Regular hexahedral intersection with four nodes in the fluid volume.
- Case 5 Irregular heptahedral intersection with four nodes in the fluid volume, the interface intersects the element at six edges, three of the sides on the fluid volume have five corners and the other three sides are triangular.
- Case 6 Same as case 3 with the fluid and inactive volumes interchanged and five nodes in the fluid volume.
- Case 7 Same as case 2 with the fluid and inactive volumes interchanged, six nodes in the fluid volume.
- Case 8 Same as case 1 with the fluid and inactive volumes interchanged, seven nodes in the fluid volume.

Figure 2 illustrates the intersections of the type of case 1 through case 6. If the finite element mesh is based on

tetrahedral pyramids only three different cases arise and the geometric setting is considerably simpler.

Figure 2 The eight possible ways a plane may intersect a hexahedron dividing it into a fluid volume where the nodes are shown, and an inactive volume (see online version for colours)



Notes: The intersections in case 3 and case 6 are of the same type, but the fluid and inactive volumes are switched around. Similarly, case 8 is obtained from case 1 and case 7 from case 2.

These are the only eight situations that may arise if the intersecting interface is a plane. The implementation is slightly more general, allowing the intersecting surface to be tri-linear in space to accommodate slight curvature effects within the element without introducing new cases. The above figure shows that the interface may intersect as many as six element edges (case 5), creating fluid volumes in the elements with shapes that are not hexahedral. To address this difficulty, two additional types of elements are allowed to be combined with the hexahedra at the interface, the linear tetrahedral pyramid in case 1 and the prismatic pentahedron in case 2. For cases 3 and 5 through 8, nodes in the inactive volume of the elements are moved in such a way as to create hexahedral elements that are a close approximation to the fluid volume geometry. The way this is done is shown in Section 4.

The use of the above intersected elements at the boundary eliminates the problem of maintaining the mesh quality and the result is a robust formulation on arbitrary geometrical configurations. At the same time, elements with

very large geometric aspects ratios may be generated; however, an analysis of the error shows that they have no detrimental effect on the accuracy of the method (Hatamipour et al., to be published). In practice, very extreme situations are easily avoided neglecting fluid elements with a volume smaller than a prescribed small fraction of the volume of the full non-intersected element, in such a way that the error introduced in the stiffness matrices is negligible.

3 Governing equations and finite element approximation

The incompressible Navier-Stokes equations, assuming zero body forces are written in non-dimensional form as

$$\frac{\partial \mathbf{U}}{\partial t} + (\mathbf{U} \cdot \nabla) \mathbf{U} = -\nabla p + \frac{1}{\text{Re}} \nabla^2 \mathbf{U} \quad (1)$$

$$\nabla \cdot \mathbf{U} = 0 \quad (2)$$

where $\mathbf{U} = (u\mathbf{i} + v\mathbf{j} + w\mathbf{k})$ is the velocity;

$$\nabla = \left(\frac{\partial}{\partial x} \mathbf{i} + \frac{\partial}{\partial y} \mathbf{j} + \frac{\partial}{\partial z} \mathbf{k} \right)$$

is the gradient operator, t is time, p is the pressure and $\text{Re} = \frac{UL}{\nu}$ is the Reynolds number. U is a characteristic velocity, L is a characteristic length and ν is the kinematic viscosity of the fluid. Equations (1) and (2) are defined over the domain $[0, T] \times \Omega(t)$ where T is a real number, and $\Omega(t)$ is a connected time dependent domain in \mathbb{R}^3 with a sufficiently smooth boundary $S(t)$.

At time $t_0 = 0$, $\Omega(t_0) = \Omega_0$ and the initial condition is $\mathbf{U}(\mathbf{x}, 0) = \mathbf{U}_0$.

The boundary conditions for each velocity component may be of the Dirichlet or Neumann type over different portions of the boundary; those portions of the boundary where Dirichlet boundary conditions are imposed are denoted by S_D and those with Neumann boundary conditions by S_N . Note that in any portion of the boundary a Dirichlet condition may be imposed on one of the velocity components and a Neumann boundary condition on another, so this notation needs to be interpreted accordingly for each velocity component. Denote $\Omega(t) \equiv \Omega_t$ at each time t ; define the space $L^2(\Omega_t)$ as the space of functions defined in Ω_t that are square integrable in Ω_t , and the space $H^1(\Omega_t)$ as the space of functions defined in Ω_t such that the function and its first partial derivatives are square integrable in Ω_t . Finally, let $S_k(t) = \{\mathbf{x}_i, i = 1, n_k / \mathbf{x}_i \in \bar{\Omega}(t)\}$ be finite sets of points that define k interfaces/boundaries contained in the reference domain that move within the domain with prescribed velocity \mathbf{v}_k . Theoretical considerations related to these algorithms have already been analysed in detail in Formaggia and Nobile (2004), Badia and Codina (2006), Guermond et al. (2006) and Boffi and Gastaldi (2004) and are not repeated here.

The ALE formulation combined with the projection method is:

- 1 Lagrangian step: Update the position of the interfaces $S_k(t)$ from time $t = t_n$ to time $t = t_{n+1} = t_n + \Delta t$ according to the prescribed velocity $\mathbf{v}_k(t)$ of each interface.
- 2 Eulerian step: Solve the Navier-Stokes equations to find $\mathbf{U}(\mathbf{x}, t_{n+1})$ and $p(\mathbf{x}, t_{n+1})$. This is done using a first order in time projection method (Quartapelle, 1993), described below.

Let $\mathbf{U}^n(\mathbf{x}) \equiv \mathbf{U}(\mathbf{x}, t_n)$ be known. At time $t = t_{n+1}$ decompose the velocity as $\mathbf{U}^{n+1} = \mathbf{U}^* + \mathbf{U}'$ where \mathbf{U}^* is an intermediate or viscous velocity that does not satisfy continuity and \mathbf{U}' is a correction or inviscid velocity that enforces the mass conservation.

To simplify the explanation, the fractional step formulation is given using only the x -component of velocity u , the equations for the other two components are similar. The time derivative is discretised using a first order backward Euler difference, the intermediate velocity component u^* is obtained solving

$$\begin{aligned} \int_{\Omega_{t_{n+1}}} \left\{ \frac{1}{\Delta t} w_u u^* + \frac{1}{\text{Re}} \nabla w_u \cdot \nabla u^* \right\} d\Omega \\ = \int_{\Omega_{t_n}} \left\{ -w_u (U^n \cdot \nabla) u^n + \frac{1}{\Delta t} w_u u^n \right\} d\Omega \end{aligned} \quad (3)$$

where w_u denotes weighting functions in $H^1(\Omega_t)$ that satisfy homogeneous Dirichlet boundary conditions in S_D . Notice that the surface integrals over the moving interfaces associated with the application of the Reynolds transport theorem to account for the time dependent domain do not appear in equation (3). This is because the velocity of the interfaces is prescribed and therefore the surface integrals vanish. If non-homogeneous Neumann conditions are applied anywhere in the domain, or if the interface velocity must be calculated as part of the solution, equation (3) must be modified by the addition of the appropriate surface integrals.

Because the convective term is kept explicit, the algorithm is subject to the Courant-Friedrich-Levy (CFL) stability condition $c \leq 1$, where c is the local Courant number. Let the subscript e denote an element, $d\Omega_e$ denote the element differential of volume and $N_i, i = 1, 8$ denote the tri-linear shape functions of the element. The full Galerkin finite element discretisation of equation (3) using eight-node isoparametric elements results in the element equations

$$\left(\frac{1}{\Delta t} \mathbf{M}^e + \mathbf{K}^e \right) (\mathbf{u}^*)^e = \mathbf{F}_u^e \quad (4)$$

where

$$\mathbf{M}^e = [m_{ij}^e] = \iiint_e N_i N_j d\Omega_e \quad (5)$$

is the element mass matrix

$$\mathbf{K}^e = [k_{ij}^e] = \frac{1}{\text{Re}} \iiint_e \left[\frac{\partial N_i}{\partial x} \frac{\partial N_j}{\partial x} + \frac{\partial N_i}{\partial y} \frac{\partial N_j}{\partial y} + \frac{\partial N_i}{\partial z} \frac{\partial N_j}{\partial z} \right] d\Omega_e \quad (6)$$

is the element stiffness matrix, $(\mathbf{u}^*)^e = (u_1^*, u_2^*, \dots, u_8^*)^T$ are the u^* component degrees of freedom contained in the element and

$$\mathbf{F}_u^e = [(f_u^e)_i] = \iiint_e N_i \left[- \left(\sum_{k=1}^8 N_k u_k^n \right) \left(\sum_{j=1}^8 \frac{\partial N_j}{\partial x} u_j^n \right) - \left(\sum_{k=1}^8 N_k v_k^n \right) \left(\sum_{j=1}^8 \frac{\partial N_j}{\partial y} u_j^n \right) - \left(\sum_{k=1}^8 N_k w_k^n \right) \left(\sum_{j=1}^8 \frac{\partial N_j}{\partial z} u_j^n \right) + \frac{1}{\Delta t} \left(\sum_{j=1}^8 N_j u_j^n \right) \right] d\Omega_e \quad (7)$$

is the forcing vector.

Two more sets of element equations,

$$\left(\frac{1}{\Delta t} \mathbf{M}^e + \mathbf{K}^e \right) (\mathbf{v}^*)^e = \mathbf{F}_v^e$$

and

$$\left(\frac{1}{\Delta t} \mathbf{M}^e + \mathbf{K}^e \right) (\mathbf{w}^*)^e = \mathbf{F}_w^e$$

for the v and w components of velocity in the y - and z -direction are obtained that differ only by the right hand sides \mathbf{F}_v^e and \mathbf{F}_w^e which are given by

$$\mathbf{F}_v^e = [(f_v^e)_i] = \iiint_e N_i \left[- \left(\sum_{k=1}^8 N_k u_k^n \right) \left(\sum_{j=1}^8 \frac{\partial N_j}{\partial x} v_j^n \right) - \left(\sum_{k=1}^8 N_k v_k^n \right) \left(\sum_{j=1}^8 \frac{\partial N_j}{\partial y} v_j^n \right) - \left(\sum_{k=1}^8 N_k w_k^n \right) \left(\sum_{j=1}^8 \frac{\partial N_j}{\partial z} v_j^n \right) + \frac{1}{\Delta t} \left(\sum_{j=1}^8 N_j v_j^n \right) \right] d\Omega_e \quad (8)$$

and

$$\mathbf{F}_w^e = [(f_w^e)_i] = \iiint_e N_i \left[- \left(\sum_{k=1}^8 N_k u_k^n \right) \left(\sum_{j=1}^8 \frac{\partial N_j}{\partial x} w_j^n \right) - \left(\sum_{k=1}^8 N_k v_k^n \right) \left(\sum_{j=1}^8 \frac{\partial N_j}{\partial y} w_j^n \right) - \left(\sum_{k=1}^8 N_k w_k^n \right) \left(\sum_{j=1}^8 \frac{\partial N_j}{\partial z} w_j^n \right) + \frac{1}{\Delta t} \left(\sum_{j=1}^8 N_j w_j^n \right) \right] d\Omega_e \quad (9)$$

respectively.

After the three assembled systems of equations have been solved for the intermediate velocity components, the pressure is obtained from the solution to the pressure Poisson equation (PPE)

$$\int_{\Omega_{t_{n+1}}} \nabla w_p \cdot \nabla p^{n+1} d\Omega = - \frac{1}{\Delta t} \int_{\Omega_{t_{n+1}}} w_p \nabla \cdot \mathbf{U}^* d\Omega \quad (10)$$

with the weighting function w_p in $H^1(\Omega_t)$. The pressure is interpolated to the same order as the velocity and the Galerkin discretisation yields the element equations

$$\mathbf{B}^e \mathbf{p}^e = \mathbf{F}_p^e \quad (11)$$

$$\mathbf{B}^e = [b_{ij}^e] = \iiint_e \left[\frac{\partial N_i}{\partial x} \frac{\partial N_j}{\partial x} + \frac{\partial N_i}{\partial y} \frac{\partial N_j}{\partial y} + \frac{\partial N_i}{\partial z} \frac{\partial N_j}{\partial z} \right] d\Omega_e \quad (12)$$

$$\mathbf{p}^e = (p_1^{n+1}, p_2^{n+1}, \dots, p_8^{n+1})^T$$

are the pressure degrees of freedom in the element and

$$\mathbf{F}_p^e = [f_i^e] = - \frac{1}{\Delta t} \iiint_e N_i \left[\sum_{j=1}^8 \frac{\partial N_j}{\partial x} u_j^* + \sum_{j=1}^8 \frac{\partial N_j}{\partial y} v_j^* + \sum_{j=1}^8 \frac{\partial N_j}{\partial z} w_j^* \right] d\Omega_e \quad (13)$$

After assembling and solving for the pressure, the velocity at time $t = t_{n+1}$ is obtained from

$$\mathbf{U}^{n+1} = \mathbf{U}^* - (\Delta t) \nabla p^{n+1} \quad (14)$$

For each velocity component at the element level, the Galerkin discretisation of equation (14) takes the form

$$\mathbf{M}^L \mathbf{u}^{n+1} = \mathbf{f}_u, \mathbf{M}^L \mathbf{v}^{n+1} = \mathbf{f}_v, \mathbf{M}^L \mathbf{w}^{n+1} = \mathbf{f}_w,$$

where \mathbf{M}^L denotes the *lumped mass matrix* (Heinrich and Pepper, 1999). The lumped mass matrices are used to avoid having to solve the extra systems of equations that would result without this modification once the global matrices are assembled. The above matrices are

$$\mathbf{M}^L = [m_i^L] = \left[\sum_{j=1}^8 \iint_e N_i N_j d\Omega_e \right] \quad (15)$$

\mathbf{M}^L is the same for the three components of velocity; \mathbf{u}^{n+1} , \mathbf{v}^{n+1} , and \mathbf{w}^{n+1} contain the eight degrees of freedom of each velocity component in the element, and the right hand sides are

$$\begin{aligned}
\mathbf{f}_u &= [(\mathbf{f}_u)_i] \\
&= \left[\sum_{j=1}^8 \iiint_e N_i N_j u_j^n d\Omega_e \right] - \Delta t \left[\iiint_e N_i \sum_{j=1}^8 \frac{\partial p_j^{n+1}}{\partial x} \right] \\
\mathbf{f}_v &= [(\mathbf{f}_v)_i] \\
&= \left[\sum_{j=1}^8 \iiint_e N_i N_j v_j^n d\Omega_e \right] - \Delta t \left[\iiint_e N_i \sum_{j=1}^8 \frac{\partial p_j^{n+1}}{\partial y} \right] \quad (16) \\
\mathbf{f}_w &= [(\mathbf{f}_w)_i] \\
&= \left[\sum_{j=1}^8 \iiint_e N_i N_j w_j^n d\Omega_e \right] - \Delta t \left[\iiint_e N_i \sum_{j=1}^8 \frac{\partial p_j^{n+1}}{\partial z} \right]
\end{aligned}$$

The systems of linear equations resulting from the assembly of equations (4) and (11) can be solved by any appropriate method, in this work, the total number of degrees of freedom involved in the simulations is rather modest (maximum 50,000 nodes) and a direct skyline method (Bathe, 1982) has been used. Obtaining the corrected velocity at time step t_{n+1} from equation (14) involves uncoupled linear systems with diagonal lumped mass coefficients matrices that are readily solved.

4 Mesh intersections and interface fitting

The reference domain Ω_0 is discretised using a mesh of isoparametric tri-linear hexahedra as shown in Figure 1(a) for a regular cubic mesh; later examples involving irregular meshes of isoparametric elements will be considered. The domain is intersected by one or more interfaces discretised using linear marker triangles in space as shown in Figure 1(b). To advance from time step t_n where the velocity and pressure are known, to time step t_{n+1} where the velocity and pressure need to be computed, an updated mesh fitted to the new position of the interface is needed. The procedure to fit the mesh to the new interface position follows the following steps:

- 1 Find all the intersections between the marker triangles and the edges of the FEM mesh elements. For the first time step, this involves a search over all elements in the mesh and all marker triangles in the interface. After the first time step, the search is narrowed to only the elements intersected at time t_n , and if a marker has moved out of an element, for this marker, the search is extended to the immediate neighbours of the element that contained the marker the previous time step. This makes the task extremely fast and efficient after the first time step.

- 2 For the first time step identify all the nodes that are in the fluid volume. To do this, the vector normal to the interface pointing toward the side occupied by "the fluid" is used; the information related to the normal to the interface is part of the input data. After the first time step, the status (fluid/inactive) of a node can be changed only if the node is an end point of an intersected element edge, adding to the efficiency of the procedure.
- 3 Modify the intersected elements to fit the interface position. How this is done for each of the different eight cases is explained below.

The simplest case is provided by the example in Figure 1 in which only intersections of the type of case 4 in Figure 2 occur; in this case, the adaptation consists in changing the location of the nodes in the inactive volume of the intersected elements to the location where the interface intersects the element's edges, as shown in Figure 3 where the repositioning of the nodes is illustrated.

At the end of these three steps, the new adapted mesh has been generated and is used to calculate velocity and pressure. There is not a unique way to complete step 3 and different strategies can be used to do it, this is easier to explain with a two-dimensional example as shown in Figure 4; where several possibilities are shown in a simple group of four bilinear elements intersected by a straight line.

A simple rectangular mesh composed of four elements and intersected by an interface is shown in Figure 4(a). Elements 1, 2 and 3 are intersected and element 4 is fully inactive. Element 1 has become a pentagon, to integrate over it, it needs to either be decomposed into a combination of quadrilateral and/or triangular elements, or modified into a single quadrilateral element. The first of these alternatives introduces additional nodes at the interface and therefore complicates the solution unnecessarily; the second option can be easily implemented by moving the inactive node number 5 to place it on the interface and modifying the geometry of the fluid elements that have node 5 in common, and this can be done in several ways. Node 5 must be relocated so as to lie on the interface, in Figure 4(b) it has been displaced toward the interface sliding along the horizontal mesh line that contains the intersection at the top of the element, in Figure 4(c), it has been displaced along the vertical mesh line intersecting the element on the right hand side, and in Figure 4(d), it has been moved to the midpoint of the interface segment intersecting element 1. All three of these options are good, and all three produce modified quadrilateral and triangular elements and it is easy to see that elements with very large aspect ratios can result; however, it can be shown that the effect of these irregularities has no detrimental effect on the accuracy of the solution; details of this analysis are discussed in Hatamipour et al. (to be published).

In three-dimensions, the same kind of geometric scenarios develop, but the effect on the neighbouring elements of displacing a node in one of them can be very complex and sometimes not possible to visualise. This suggests that a fourth option may be considered illustrated in Figure 4(e) where each of the intersected elements is modified individually, hence, when processing element 1 it is modified displacing node 5 as in Figure 4(d), in element 2 node 5 is moved as in Figure 4(b), and in element 3 node 5 is moved as in Figure 4(c). This has the advantage that the shape of the elements remains more uniform, and therefore the approximation error is reduced (Strang and Fix, 1973), on the other hand it has the disadvantage that it introduces a

local inconformity because small areas may be left out in the integrations, these are the shaded areas shown in Figure 4(e); however, this amounts to an approximation of the domain boundary which is $O(h^3)$, the lack of conformity does not violate the patch test and the error remains of order $O(h^2)$ where h is the mesh parameter (Strang and Fix, 1973). In three-dimensions, the last option makes it possible to visualise the deformed geometries for every situation, this has not been possible in the other cases, it also prevents the fitted elements from becoming excessively deformed. The different options have all been implemented and numerical simulations have shown no discernible difference in the results.

Figure 3 Mesh adaptation used to generate the modified mesh in Figure 1

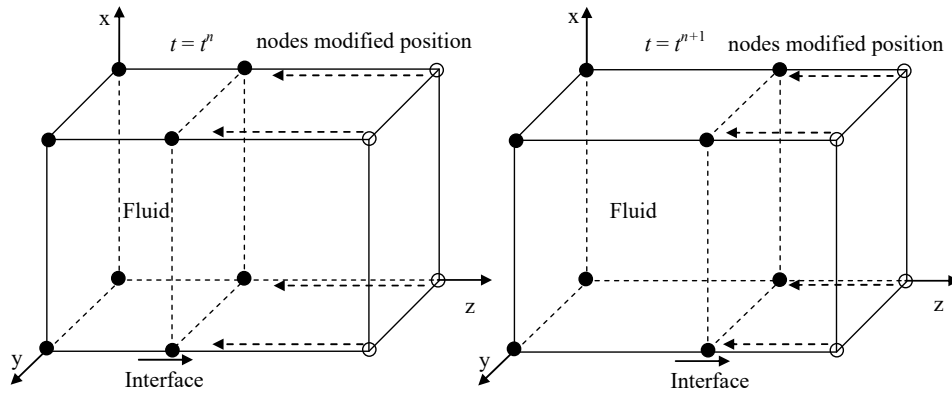


Figure 4 Two-dimensional example to illustrate different mesh adaptation strategies, (a) four element mesh intersected by an interface (b) inactive node 5 moved along a horizontal mesh line (c) inactive node 5 moved along a vertical mesh line (d) inactive node 5 moved to the centre of the intersection segment (e) a combination of the three previous cases (see online version for colours)

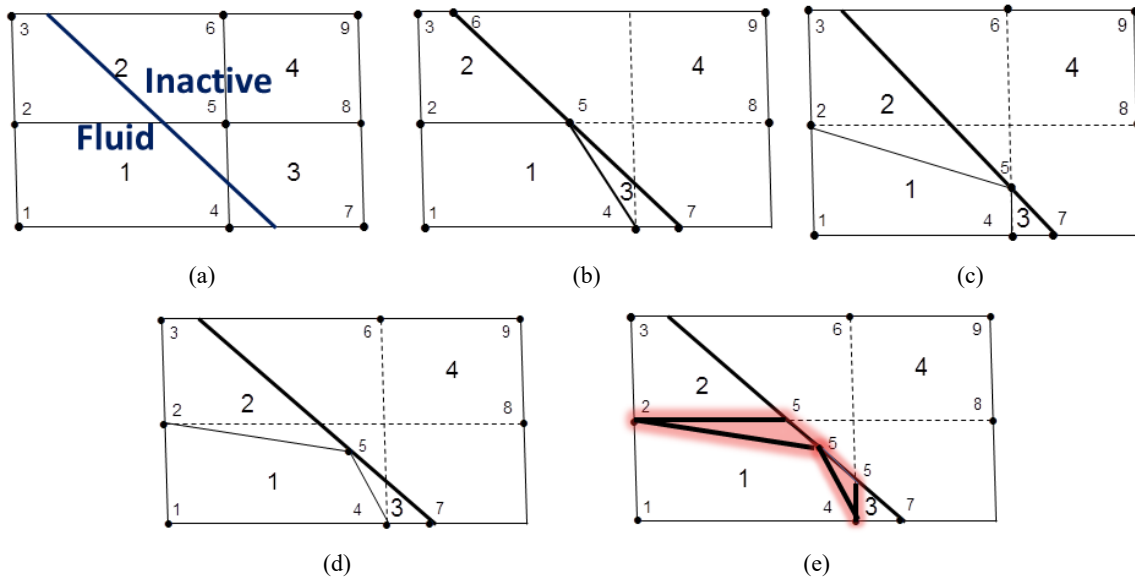
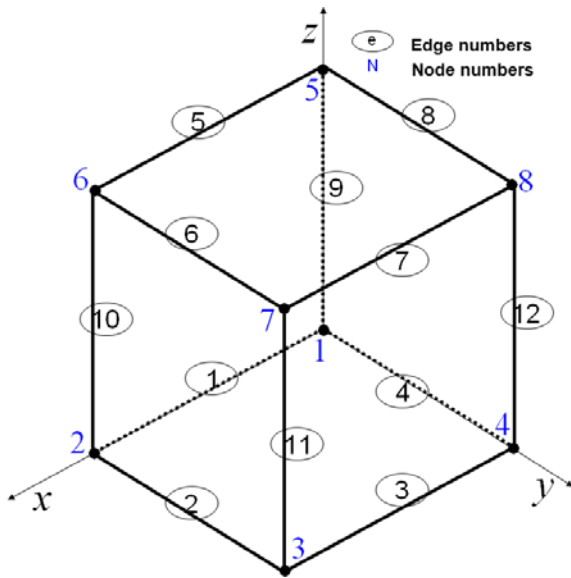


Figure 5 Notation used to describe the element edges and nodes (see online version for colours)



To explain how the adaptation process described above in two-dimensions is implemented in three-dimensions the notation in Figure 5 is used to number the nodes and edges of a tri-linear element.

For element intersections in cases 1 or 2 of Figure 2, the fluid volume is a pyramid or prismatic element respectively and is integrated using isoparametric coordinates. When an element is of the type of case 3, it is modified into an isoparametric hexahedron as shown in Figure 6 in the following way: First, the element is rotated to the standard position shown in Figure 6(a), where nodes 1, 2 and 4 are the fluid nodes. Next, the inactive nodes are repositioned as shown in Figure 6(b), so that node 3 is moved to the midpoint of the segment connecting the intersections on edges 2 and 3; node 5 is moved to the intersection point in edge 9; node 6 is moved to the intersection point in edge 10; node 7 moves to the midpoint between the repositioned nodes 3 and 5 and node 8 is moved to the intersection point on edge 12. The end result is the degenerate hexahedron shown in Figure 6(c), in which nodes 3, 5, 6, 7 and 8 all lie in the interface. However, the integrations over the volume, and the volumes that are generated by cases 5 and 6 as well, are automatically done using the hexahedral shape functions in conjunction with isoparametric transformations.

Figure 6 (a) Element intersection of the type of case 3 (b) Displaced position of nodes 3, 5, 6, 7 and 8 (c) Final modified geometry (see online version for colours)

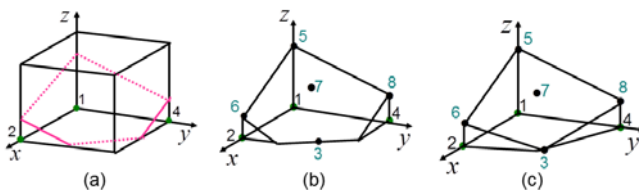


Figure 7 (a) Element intersection of the type of case 5 (b) Displaced position of nodes 3, 6, 7 and 8 (c) Final modified geometry (see online version for colours)

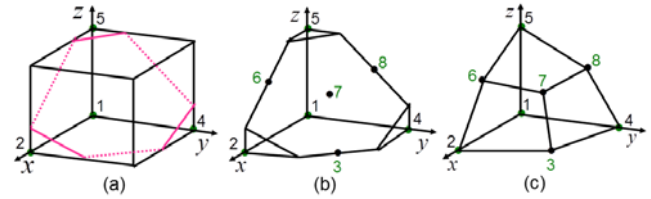


Figure 8 (a) Element intersection of the type of case 6 (b) Displaced position of nodes 6, 7 and 8 (c) Final modified geometry (see online version for colours)

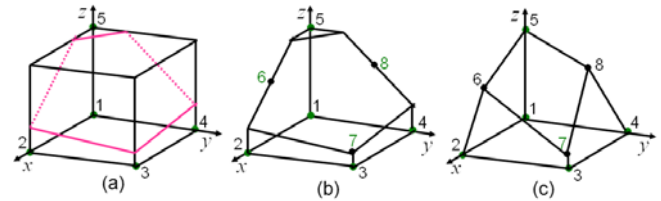


Figure 9 (a) Element intersection of the type of case 7 (b) Displaced position of nodes 7 and 8 (c) Final modified geometry (see online version for colours)

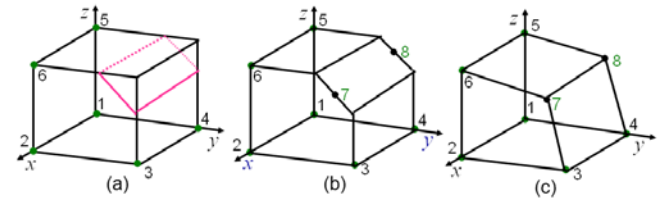
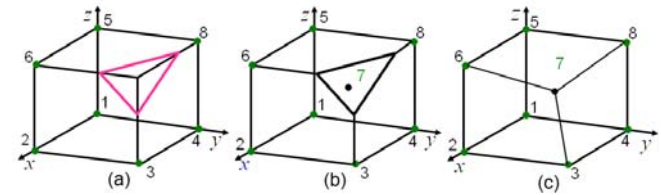


Figure 10 (a) Element intersection of the type of case 8 (b) displaced position of nodes 7 (c) Final modified geometry (see online version for colours)



Element intersections of the type of case 4 result in standard isoparametric hexahedrons that do not require any special treatment. Intersections of the type of case 5 follow the steps shown in Figure 7; the element is rotated to the standard position shown in Figure 7(a); where nodes 1, 2, 4 and 5 are in the fluid. In Figure 7(b), the inactive nodes are repositioned, node 3 is moved to the midpoint of the segment connecting the intersections on edges 2 and 3; node 6 moves to the midpoint of the segment connecting the intersections on edges 5 and 10; node 8 is moved to the midpoint of the segment connecting the intersections on edges 8 and 12 and node 7 is placed at the centroid of nodes 3, 6 and 8. The nodes 3, 6, 7 and 8 are now all on the interface. The modified element is shown in Figure 7(c).

An element intersection of the type of case 6 is shown in Figure 8(a). The element is rotated to a position where nodes 1, 2, 3, 4 and 5 are in the fluid. In Figure 8(b), the inactive nodes are repositioned, node 6 is on the midpoint of the segment connecting the intersections on edges 5 and 10; node 7 has been displaced along edge 11 to the intersection point on that edge, and node 8 was moved to the midpoint of the segment connecting the intersections on edges 8 and 12. Figure 8(c) shows the final configuration for the modified element and 8 are the inactive nodes. In Figure 9(b), the displaced position of nodes 7 and 8 to the midpoints of the segments connecting the intersected edges 6 and 11 and edges 8 and 12 respectively is shown, and Figure 9(c) has the final modified geometry.

Finally, an element intersection of the type of case 8 is shown in Figure 10(a) rotated so that the only inactive node is node 7. The modified location of node 7 is shown in Figure 10(b) and is the centroid of the triangle determined by the intersections on edges 6, 7 and 11. The final configuration is as in Figure 10(c).

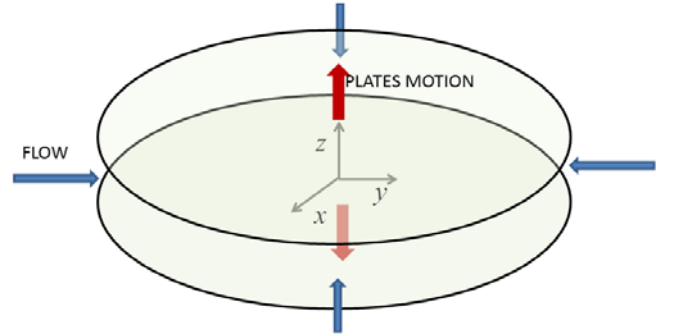
The modification of the intersected elements completes the adaptation process at the current position of the interface. As was explained at the beginning of the section, there is not a unique way to do this, because the nodes are typically common to eight adjacent elements and, as seen in Figure 4(e), depending on the order the element are processed the nodes will be moved to different new locations, therefore some decisions have to be made as to how the adaptation will take place. Three different approaches have been explored in this work, in the first priority is given to sliding the nodes along the intersected edges rather than moving them to a midpoint between intersections. Therefore, once a node has been displaced along an edge it is not allowed to be moved again when considering the other elements that contain it. On the other hand, if the node has been moved to a midpoint between intersections in an element and later a neighbouring element requires the node to slide along an intersected edge, the first displacement is discarded, the node is moved according to the current requirement and there it is fixed and not moved again regardless of repositioning requests coming from other elements later. The second approach is the opposite of the first; it gives priority to repositioning the nodes at midpoints over sliding along edges. Both of these approaches result in perfectly conformal adaptations that cover the entire fluid volume and do not have overlaps; however, it is not possible to visualise some of these final configurations and some much distorted elements can be generated. For this reason, the approach explained in two-dimensions through Figure 4(e), in which the elements are modified independently has also been implemented. The drawback is that as shown in the figures the final adaptation does not cover the entire fluid domain, small errors in the volume are introduced and the integrals are not exact, it has been proved (Strang and Fix, 1973) in the two-dimensional case that the additional error due to inaccuracies in the approximation of the boundary geometry is $O(h^3)$, where h characterises the size of the mesh, referred to as the mesh

parameter, and therefore the approximation error remains $O(h^2)$; the error introduced in this case also amounts to an additional local distortion of the boundary, and the analysis must be valid for this case also. It is also well-known that the constant of proportionality in the error associated with an isoparametric transformation is directly proportional to the maximum value of the Jacobian of the transformation and to the product of the norms of the transformation and its inverse, as well as inversely proportional to the minimum value of the Jacobian of the transformation (Strang and Fix, 1973). As a consequence approximation errors are greatly increased as the elements become much distorted. Which of the three alternatives is better cannot be assessed and is also problem dependent. So far, simulations performed using all three approaches show no significant differences in the velocity components next to the interface.

5 Model validation

To assess the accuracy and general behaviour of the simulations in the present model the unsteady flow of a viscous fluid between two parallel circular plates that are separating at a prescribed velocity is considered, for which analytical solutions have been obtained (Wang, 1976; Riley and Drazin, 2006). Figure 11 depicts the physical arrangement.

Figure 11 Fluid layer between circular plates separating at a prescribed velocity (see online version for colours)



If the position and velocity of the upper and lower interfaces are given by $h(t) = \pm h_0(1 - \alpha t)^{1/2}$ and $w_h(t) = \mp \alpha h_0 / 2(1 - \alpha t)^{1/2}$ respectively, where $h_0 \equiv h(0)$ is the initial position of the interface, a similarity solution exists for the velocity field of the form

$$u(x, y, z, t) = \frac{\alpha x}{4(1 - \alpha t)} f'(\eta) \quad (17)$$

$$v(x, y, z, t) = \frac{\alpha y}{4(1 - \alpha t)^{1/2}} f'(\eta) \quad (18)$$

$$w(x, y, z, t) = \frac{-\alpha h_0}{2(1 - \alpha t)^{1/2}} f(\eta) \quad (19)$$

where $\eta = \frac{z}{h_0(1-\alpha t)^{1/2}}$ is a stretched vertical coordinate, and f is the solution to the ordinary differential equation

$$\eta f''' + 3f'' - f f'' = \frac{1}{S} f'''' , \quad (20)$$

$$f(0) = f'(1) = f''(0) = 0, f(1) = 1$$

$S = \alpha h_0^2 / 2\nu$ is a Reynolds number; ν is the kinematic viscosity of the fluid; α is a parameter with units of $1/\text{time}$, with $\alpha > 0$ for squeezing flow and $\alpha < 0$ for expanding flow.

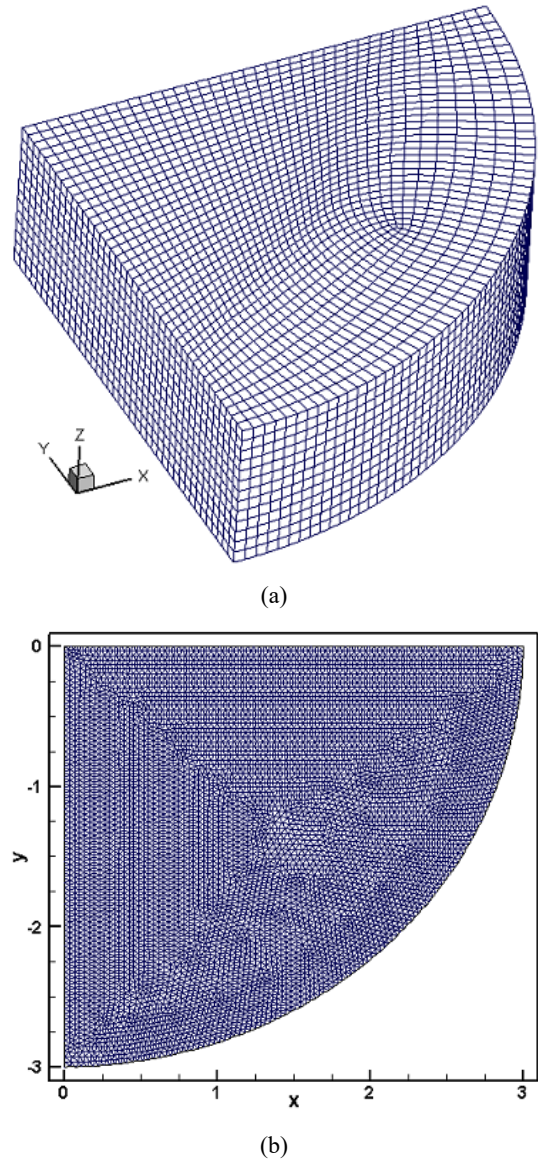
Solution for squeezing and expanding flows over a range of Reynolds numbers are presented in Wang (1976). However, it has only been possible to obtain numerical solutions of the flow field that are in agreement with the analytical ones in the case of Stokes flow and for $\alpha < 0$. To find an analytical solution for the pressure field is much more difficult due to the appearance of a time dependent constant in the integration of the equations that cannot be properly determined. Under additional assumptions, it is possible to obtain an expression for the pressure with an error $O(h_0 / R)$, where R is the finite radius of the plates (Riley and Drazin, 2006); however, the numerical solutions obtained for the pressure do not approximate the resulting analytical solution obtained using the time dependent constant. Therefore, results presented here are limited to an assessment of the accuracy of the flow field and conditions of Stokes flow and $\alpha < 0$. In this case, the solution to equation (20) is

$$f(\eta) = \frac{3}{2}\eta - \frac{1}{2}\eta^3 \quad (21)$$

The results for the case when $\alpha = -1.5$, $h_0 = 0.4$ and domain radius $R = 3$ are presented here; only the upper half of the domain $z \geq 0$ is considered with symmetry boundary conditions at $z = 0$. The numerical simulations are performed in a domain consisting of the 3D quadrant $x > 0$, $y < 0$ and symmetry conditions are imposed along the planes $x = 0$ and $y = 0$. Figure 12(a) depicts the reference domain Ω_T that in the z -direction spans over $0 \leq z \leq 1$ discretised with one of the finite element meshes used in the simulations; the marker triangles grid that describes the moving planar interface used with this finite elements mesh is shown in Figure 12(b).

The boundary conditions on the velocity field are the exact analytical solution at the outer radius $R = 3$; on the moving interface $u = v = 0$ and $w_h(t) = -\alpha h_0 / 2(1 - \alpha t)^{1/2}$; along the plane $y = 0$ the boundary conditions are $v = \partial u / \partial y = \partial w / \partial y = 0$; along the plane $x = 0$, $u = \partial v / \partial x = \partial w / \partial x = 0$ and at the bottom of the domain $z = 0$ the boundary conditions are $w = \partial u / \partial z = \partial v / \partial z = 0$. For the pressure, the normal derivative at the boundaries is set to zero everywhere and the pressure is fixed at one point in the domain; this point will be identified later as the details of the finite element meshes are described.

Figure 12 Comparison problem, circular plates moving away at a prescribed velocity, (a) domain and computational mesh (b) moving top boundary interface (see online version for colours)

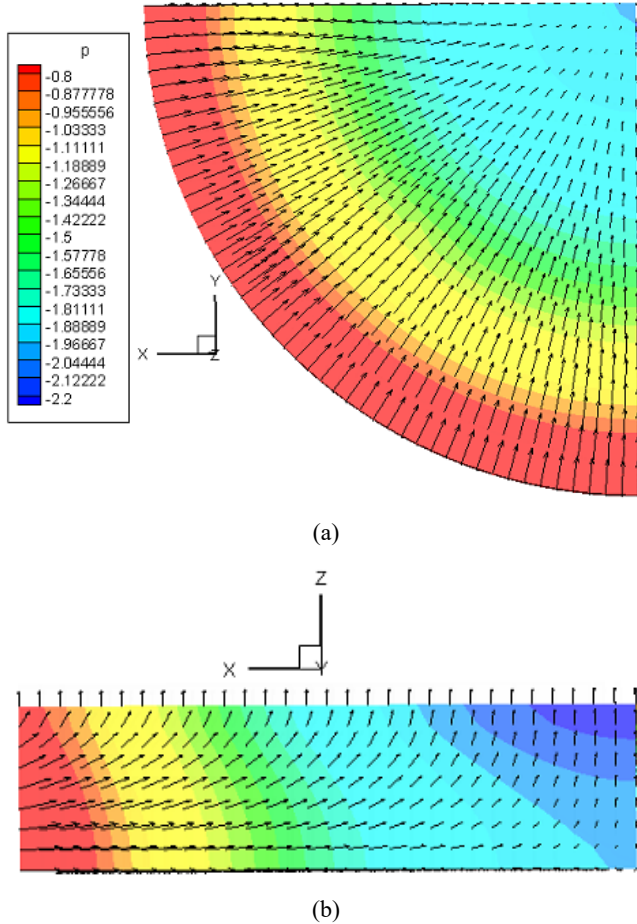


All the computational meshes and the discretisation of the moving interfaces in this work have been produced using the program CUBIT developed at the Sandia National Laboratory (The CUBIT Geometry and Mesh Generation Toolkit, 2014). For the purpose of comparison of the numerical and analytical solutions three meshes are used, with the elements generated as close as possible to uniform cubes as shown in Figure 12(a). The coarsest mesh, denoted by mesh 1 consists of 5,690 hexahedral elements and 6,798 nodes, the element sizes are as close as possible to regular cubes of size $h = 0.1$. For this mesh, the top moving interface is discretised using 2,276 marker points and 4,372 marker triangles. The second mesh consists of hexahedra as close to cubes of size $h = 0.075$ as possible, it is denoted as mesh 2, and together with the top interface discretisation is the one depicted in Figure 12. The finite element mesh has

19,968 elements and 22,526 nodes, and the top interface is described by 6,057 markers and 11,824 marker triangles. The size of the finest mesh, mesh 3 is very close to one half that of mesh 1, or $h = 0.05$, it has 45,520 hexahedral elements and 49,833 nodes. The top interface is described using 9,965 markers and 19,544 marker triangles.

The initial condition is the exact velocity, in meshes 1 and 3 the pressure is fixed at the node located at the position (1.3333, -1.3405, -0.3), and in mesh 2 the pressure is fixed at (1.3274, -1.3274, -0.3462).

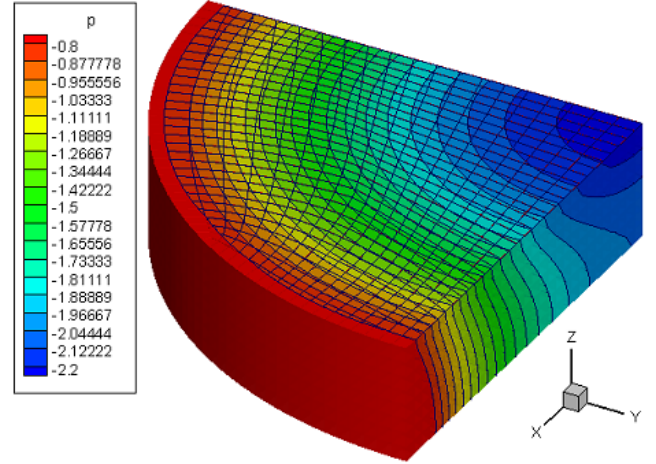
Figure 13 (a) Calculated velocity field and pressure in the $x - y$ plane at $z = 0$ and time $t = 2.0$ (b) Calculated velocity field and pressure in the plane $y = 0$ at $t = 2.0$ (see online version for colours)



Simulations utilising the three meshes were done on the time domain $[0, 3]$. The initial position of the top interface is $h_0 = 0.4$ and the final position is $h(3) = 0.938$. Velocity and pressure obtained with mesh 1 at time $t = 2$, when the interface is at $h = 0.8$ is shown in Figure 13. It can be observed that qualitatively the flow exhibits the expected features, it is perfectly axisymmetric and it turns from parallel to the $x - y$ plane at $z = 0$ to perpendicular to it at the top interface. The pressure at time $t = 2$ at the top interface and the lateral boundaries where $x > 0$ is shown in more detail in Figure 14. The pressure also shows a perfectly axisymmetric distribution. Moreover, as we will see below, the velocity field is being captured with excellent accuracy and exhibits a second order convergence rate when

compared with the solution in equations (17) to (19) and (21), therefore, there is no reason to believe that the numerical solution obtained here for the pressure field is not also accurate. In two-dimensions, a solution to the flow and pressure between two parallel plates separating with a prescribed velocity can be found and shows that the pressure is accurate and exhibits a convergence rate in the neighbourhood of 1.5, the details are presented in Hatamipour et al. (to be published).

Figure 14 Pressure field at $t = 2$, results obtained with mesh 1 (see online version for colours)



To assess the accuracy of the present method consider three calculations, one each on meshes 1, 2 and 3. In the time interval $0 \leq t \leq 3$. The time step is chosen as $\Delta t = 0.01$ for mesh 1, $\Delta t = 0.00563$ for mesh 2 and $\Delta t = 0.0025$ for mesh 3 to make the error introduced by the first order time stepping scheme consistent with that of a second order accurate scheme in time. The results of the simulations are compared directly with the analytical solution calculating the relative errors in the velocity.

It is well-known that meaningful comparisons of numerical and analytical solutions can be very difficult to perform, especially in a three-dimensional problem with non-rectangular domain. Here, the relative error in the numerical solution is measured at every active node for the total velocity magnitude and each of the three velocity components, and then they are averaged over the whole domain to obtain a measure of the relative error at each time step. A more detailed discussion of the error measurement in this problem is given in Hatamipour et al. (to be published). The average relative error in the velocity magnitude at each time step t_n is defined as

$$E(t_n) = \left(\frac{\sum_{i=1}^k [(u_i^* - u_i^n)^2 + (v_i^* - v_i^n)^2 + (w_i^* - w_i^n)^2]}{\sum_{i=1}^k [(u_i^*)^2 + (v_i^*)^2 + (w_i^*)^2]} \right)^{1/2} \quad (22)$$

The average relative errors for the velocity magnitude are shown in Figure 16 as a function of time. The individual velocity components all show the same behaviour and are

not presented here. Figure 15 shows that the results are accurate and converge as the mesh is refined. To assess the order of convergence, the time dependent average errors in Figure 15 are now averaged over the time interval $0 \leq t \leq 3$, this produces one number that is representative of the total error in each simulation. The expressions for the total averaged relative error is $(\bar{E})_j = \frac{1}{2} \int_0^2 E_j(t) dt$ where $j = 1, 2, 3$ denotes each of the meshes. These numbers are plotted against the corresponding mesh size in a log-log scale in Figure 16, together with their linear least-squares fit, and are also listed in Table 1. The slope of the linear least-squares fit is the convergence rate derived from this problem and in this case it is 1.901, that is, the convergence is of second order in space.

Figure 15 Relative error in the velocity magnitude for meshes 1, 2 and 3 (see online version for colours)

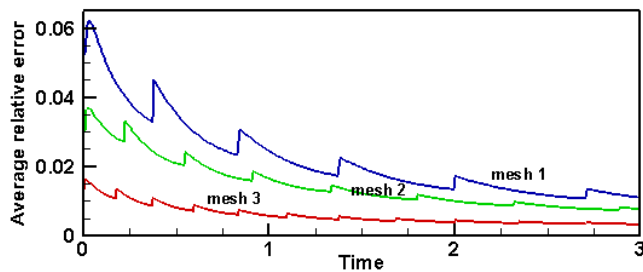
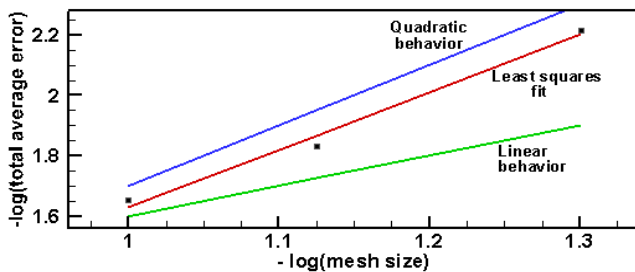


Figure 16 Convergence rate for the velocity magnitude in flow between parallel plates separating with prescribed velocity (see online version for colours)



Note: The calculated convergence rate is 1.901.

Table 1 Total averaged spatial relative error for the separating disks problem

j	h_j	$(\bar{E})_j$
1	0.1	0.02235
2	0.075	0.01475
3	0.05	0.00607

Clearly, this one example is not proof of second order convergence; however, it gives a strong indication of the method's behaviour and it has been observed to be consistent on other similar calculations. The same behaviour has been observed in two-dimensions and a more systematic local analysis presented in detail in Hatamipour et al. (to be published) shows the error to be consistently of second order.

6 Further examples

To illustrate the capabilities of the model two additional examples of pistons moving in an engine cylinder are given, that involve different types of piston geometry.

6.1 Engine cylinder with cylindrical bowl piston

The computational geometry consists of a cylindrical domain representing an engine cylinder with two valve openings for intake and exhaust of gases, and a piston with a cylindrical bowl. The cylinder has an inner radius of 5 cm and a height of 10.5 cm; the piston has a planar ledge with an outer radius of 5 cm and an inner radius of 3.75 cm where a straight cylinder 1.5 cm tall is attached going downward and capped by a plane at the bottom. The cylinder is discretised with a mesh of 7,152 nodes and 6,150 tri-linear isoparametric elements and the piston is defined by a grid of linear triangles containing 5,104 markers and 10,062 marker triangles. Figure 17 shows the piston/cylinder geometries. The cylinder mesh consists of 15 uniformly spaced rows of elements parallel to the $x - y$ plane; each plane of nodes contains 410 elements. The mesh has been chosen coarse enough to show the effectiveness of the algorithm while keeping the visualisation reasonably clear.

The simulation presented here assumes that the fluid is air, but speeds are in the incompressible range, the reference length and velocity are set to 1.0 and the kinematic viscosity is $\nu = 1.7 \times 10^{-3}$ which results in a Reynolds number $Re = 588$. The top surface of the piston is initially at $z = 3.25$ cm and it reaches a maximum height of $z = 11.0$ cm, the amplitude of the stroke is 7.75 cm. The piston is driven according to the function $h(x, y, t) = h(x, y, 0) + 0.03875(1 + \sin(50\pi t - \pi/2))$ where h is in metres, so the maximum velocity of the piston when $t = 0.1 + 0.02n$, $n = 0, 1, 2, \dots$ is $w_p = 6.087$ m/s, and a whole cycle requires 0.04 s for completion, a rotation rate of 1,500 rpm. Figure 18 shows four snapshots during one full cycle of motion.

At $t = 0.005$ s the piston has advanced 0.113 cm and the piston velocity at this point is 4.3 m/s, the exhaust valve is open and fluid leaving the cylinder; at $t = 0.015$ s it is approaching the top, the exhaust valve is still open the piston top is at $z = 9.86$ cm and its velocity is 4.3 m/s; at $t = 0.025$ s the piston has turned, its position is back to $z = 9.86$ cm, and its velocity is now -4.3 m/s; finally, at $t = 0.035$ s the piston approaches the bottom of the cylinder to start a new cycle, its position is back to $z = 4.38$ cm and its velocity is 4.26 m/s. A uniform time step of 10^{-5} s was used in the simulation and a full cycle required an average 0.48 seconds CPU per time step in a Dell T5600 workstation.

Figure 17 (a) Two-dimensional horizontal cross-section of the cylinder mesh showing the circular intake and exhaust valve openings (b) Cylinder/piston configuration showing the outer surfaces of the cylinder mesh (see online version for colours)

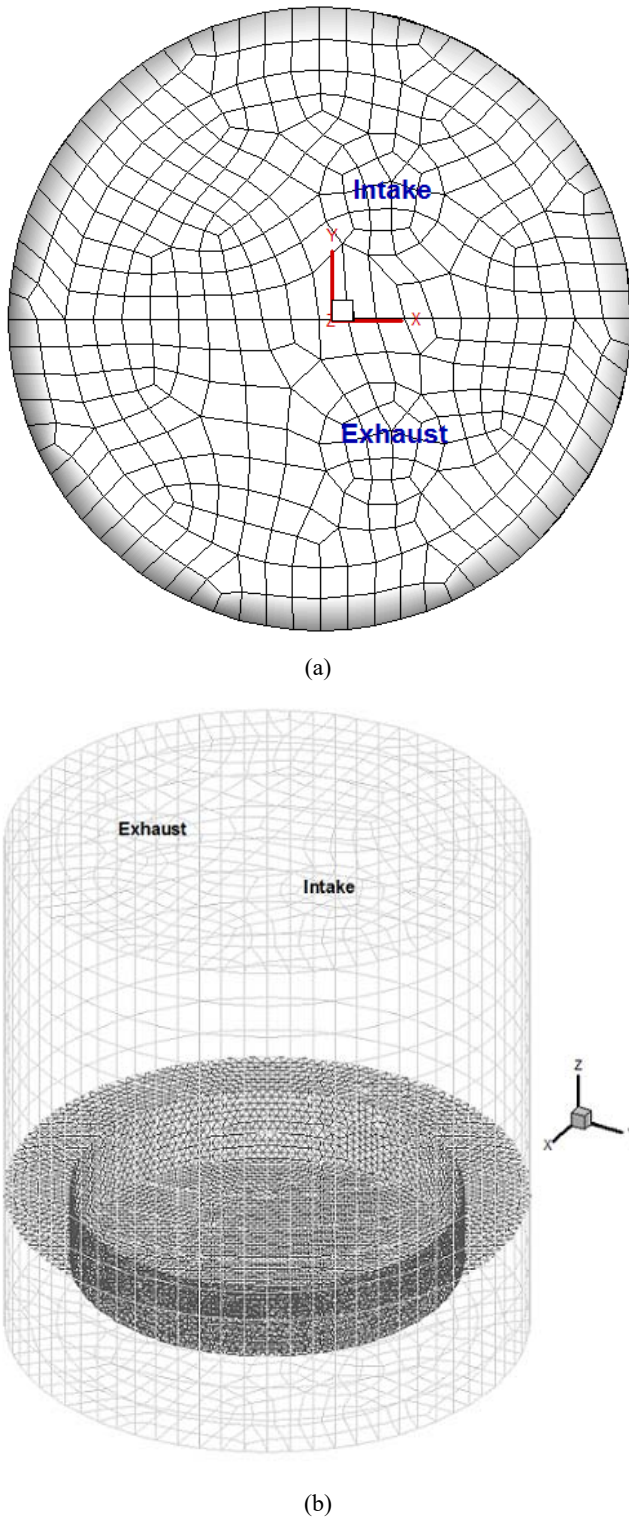
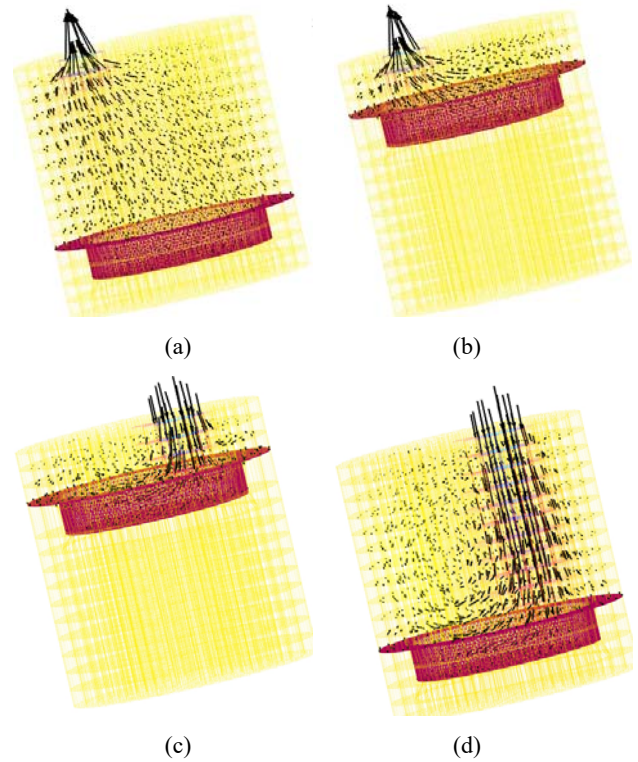


Figure 18 (a) Piston position and velocity field at $t = 0.005$ s (b) Piston position and velocity field at $t = 0.015$ s (c) Piston position and velocity field at $t = 0.025$ s (d) Piston position and velocity field at $t = 0.035$ s (see online version for colours)



6.2 Engine cylinder with curved bowl piston

The second example deals with a piston head geometry representative of today's advanced designs consisting of a curved bowl geometry shown in Figure 19 together with the detailed geometry dimensions of the piston head. The inner diameter of the piston and cylinder is 10 cm. The piston head is described with a mesh that contains 4,681 marker points and 9,236 marker triangles. The cylinder and finite element mesh are shown in Figure 24; the cylinder is 10.5 cm tall and is discretised by 33 planes of elements parallel to the z -axis containing 437 elements each as shown in Figure 20. The mesh contains a total of 14,421 hexahedral elements and 15,946 nodes.

The top surface of the piston is initially at $z = 2.2$ cm from the bottom of the cylinder and it reaches a maximum height of $z = 9.44$ cm, the amplitude of the stroke is 7.24 cm. The piston is driven according to the function $h(x, y, t) = h(x, y, 0) + 0.0362(1 + \sin(30\pi t - \pi/2))$, where h is in metres, the maximum velocity of the piston is $w_p = 3.412$ m/s. A typical time step in these simulations is $\Delta t = 0.0002$, a full cycle takes 0.067 s and is completed in 333 time steps. Figure 21 shows snapshots of a full cycle at six different times.

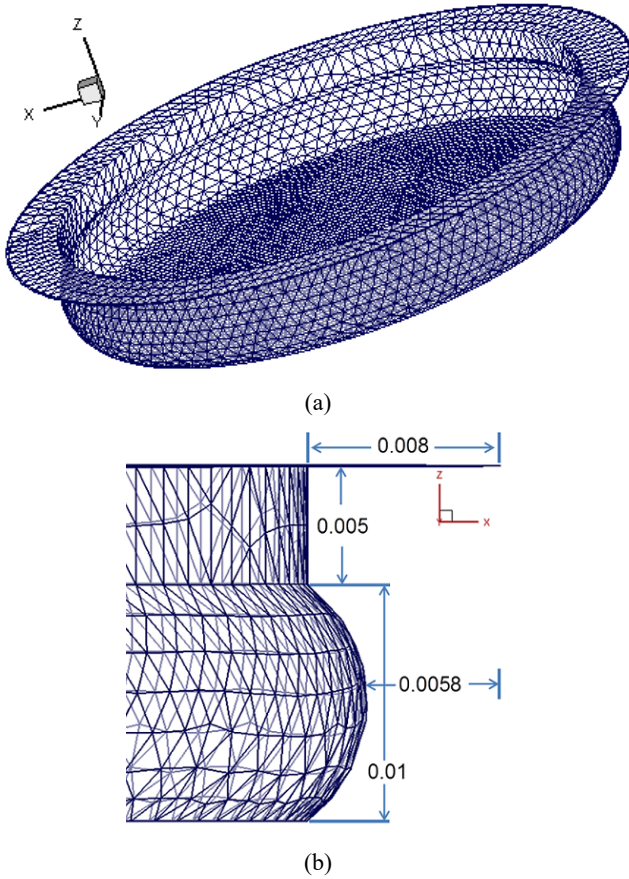
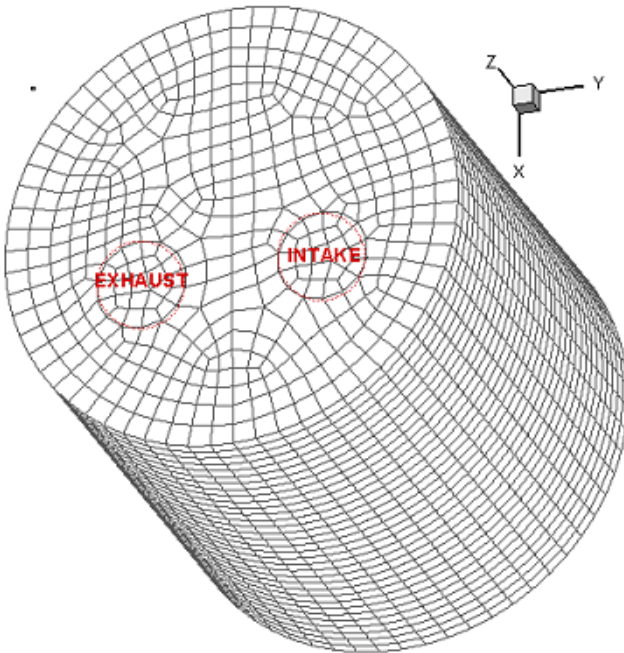
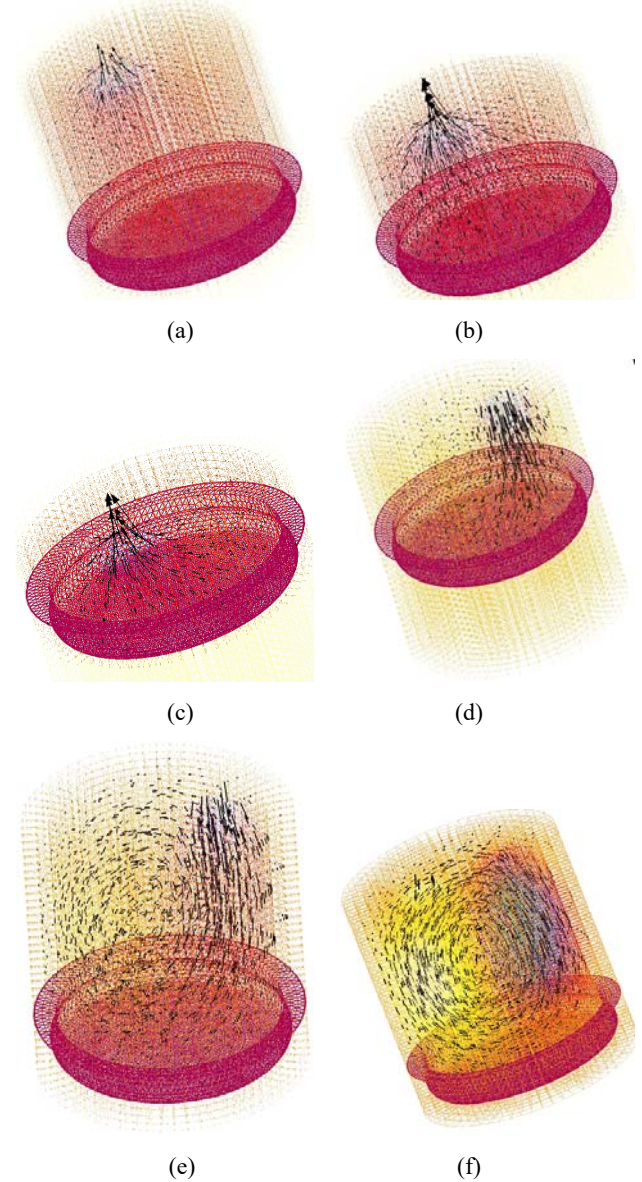
Figure 19 (a) curved bowl piston head geometry (b) Piston head dimensions (see online version for colours)**Figure 20** Curved bowl cylinder and finite element mesh (see online version for colours)**Figure 21** Flow inside the bowl cylinder (a) piston position and velocity field at $t = 0.005$ s (b) piston position and velocity field at $t = 0.015$ s (c) piston position and velocity field at $t = 0.025$ s (d) piston position and velocity field at $t = 0.05$ s (e) piston position and velocity field at $t = 0.06$ s (f) piston position and velocity field at $t = 0.067$ s (see online version for colours)

Figure 21(a) shows the piston position and velocity field at time $t = 0.005$ s as the piston starts moving up from rest. The top of the interface is at $z = 2.59$ cm and the velocity of the piston is $w_p = 1.55$ m/s. In Figure 21(b), the time is $t = 0.015$ s and the interface has reached $z = 5.25$ cm, the velocity of the piston is close to its maximum at $w_p = 3.37$ m/s. Figure 21(c) shows time $t = 0.025$ s the piston is close to the maximum height at $z = 8.38$ cm and its velocity is $w_p = 2.41$ m/s. In Figure 21(d), at time $t = 0.05$ s the piston is on its way down, the exhaust valve has been closed and the intake valve is now open; the interface position is at $z = 5.82$ cm and the velocity of the piston is $w_p = -3.41$ m/s, very near its minimum. Figure 21(e) shows the piston and fluid field at time $t = 0.06$ s as the completion

of the cycle is approached, the piston is at $z = 2.76$ cm and its velocity is $w_p = -2.01$ m/s. Finally, Figure 21(f) depicts the situation at $t = 0.067$ s, this is 0.002s after the piston has completed the cycle and started moving back up again. The top of the interface is at $z = 2.202$ cm and the velocity of the piston is $w_p = 0.107$ m/s. The intake valve has been closed and the exhaust valve is open, notice that now the simulation does not proceed from conditions of fluid at rest and the flow field has not come to a full stop so the second and subsequent cycles have somewhat different dynamics. During the simulation encompassing, the first full cycle the magnitude of the fluid's vertical velocity w varied between $-90.1 \text{ m/s} \leq w \leq 50.3 \text{ m/s}$.

7 Conclusions

In this work, a method based on finite element discretisation that includes moving boundaries or interfaces in CFD calculations has been developed and implemented in three space dimensions. The method belongs to the general family of ALE methods. But differs from previously proposed methods for similar problems in the way the computational mesh is defined and locally fitted as the geometry changes. The present work concentrates strictly on the development of the method to discretise and modify the mesh throughout the calculation, and example simulations have been presented only for laminar incompressible flow at low Reynolds number. The same methodology has been extended to high Reynolds number compressible and turbulent flows and will be reported in the future. The method has been fully tested for accuracy in two- and three-dimensions and shown to have second order accuracy in space. Additional details of error analysis are presented in the second part of this work (Hatamipour et al., to be published).

The simulations presented here have been performed in two ways, first restricted to just the interface motion, without solving for the velocity and pressure; then solving the complete fluid problem, and the total CPU times for the two calculations have been compared. The results show that in all cases the work necessary to simulate the interface motion amounts to less than 1% of the total simulation time, and confirm that the method presented here is extremely efficient. Furthermore, the examples involving simple interfaces and those involving a more realistic cylinder/piston engine assembly show that the method is fully robust as well as very efficient, and that it can offer distinct advantages over other existing methods from the point of view of eliminating the need to use adaptive mesh generation.

Acknowledgements

The DOE's Office of Energy Efficiency and Renewable Energy (EERE) Advanced Combustion Program (Gurpreet Singh) is supporting this effort. Los Alamos National Laboratory, an affirmative action/equal opportunity

employer, is operated by the Los Alamos National Security, LLC for the National Nuclear Security Administration of the U.S. Department of Energy (DOE) under contract DE-AC52-06NA25396. Los Alamos National Laboratory strongly supports academic freedom and a researcher's right to publish; as an institution, however, the Laboratory does not endorse the viewpoint of a publication or guarantee its technical correctness.

References

- Askes, H. and Sluys, L.J. (2000) 'Remeshing strategies for adaptive ALE analysis of strain localization', *Eur. J. Mech. A/Solids*, Vol. 19, No. 3, pp.447–467.
- Badia, S. and Codina, R. (2006) 'Analysis of a stabilized finite element approximation of the transient convection-diffusion equation using an ALE framework', *SIAM J. Numer. Anal.*, Vol. 44, No. 5, pp.2159–2197.
- Bathe, K.-J. (1982) *Finite Element Procedures in Engineering Analysis*, Prentice-Hall, Englewood Cliffs, New Jersey.
- Boffi, D. and Gastaldi, L. (2004) 'Stability and geometric conservation laws for ALE formulations', *Comp. Meth. Appl. Mech. Engng.*, Vol. 193, No. 42, pp.4717–4739.
- Brackbill, J.U. and Saltzman, J.S. (1982) 'Adaptive zoning for singular problems in two dimensions', *J. Comp. Phys.*, Vol. 46, No. 3, pp.342–368.
- Carrington, D.B. (2011) *A Fractional Step HP-Adaptive Finite Element Method for Turbulent Reactive Flow*, Los Alamos National Laboratory Report, LA-UR-11-00466.
- Carrington, D.B., Muñoz, D.A. and Heinrich, J.C. (2014a) 'Modelling fluid flow in domains containing moving interfaces', *Progress in Computational Fluid Dynamics*, Vol. 14, No. 3, pp.139–150.
- Carrington, D.B., Wang, X. and Pepper, D.W. (2014b) 'A predictor-corrector split projection method for turbulent reactive flow', *Journal of Computational Thermal Sciences*, Vol. 5, No. 4, pp.333–352, Begell House Inc.
- Codina, R., Houzeaux, G., Coppola-Owen, H. and Baiges, J. (2009) 'The fixed-mesh ALE approach for the numerical approximation of flows in moving domains', *J. Comp. Phys.*, Vol. 228, No. 5, pp.1591–1611.
- Dettmer, W.G. and Perić, D. (1996) 'A fully implicit computational strategy for strongly coupled fluid solid interaction', *Arch. Comput. Methods Eng.*, Vol. 14, No. 3, pp.205–247.
- Donea, J., Giuliani, S. and Halleux, J. (1982) 'An arbitrary Lagrangian-Eulerian finite element method for transient dynamic fluid-structure interactions', *Comp. Methods Appl. Mech. Eng.*, Vol. 33, No. 1, pp.689–723.
- Farhat, C. and Geuzaine, P. (2004) 'Design and analysis of robust ALE time-integrators for the solution of unsteady flow problems on moving grids', *Comp. Methods Appl. Mech. Engng.*, Vol. 193, No. 39, pp.4073–4095.
- Formaggia, L. and Nobile, F. (2004) 'Stability analysis of second order time accurate schemes for ALE-FEM computations', *Comp. Meth. Appl. Mech. Engng.*, Vol. 193, No. 39, pp.4097–4116.
- Gadala, M.S. (2004) 'Recent trends in ALE formulations and its applications in solid mechanics', *Comp. Methods Appl. Mech. Engng.*, Vol. 193, No. 1, pp.4247–4275.

- Guermond, J.L., Mineev, P. and Shen, J. (2006) 'An overview of projection methods for incompressible flow', *Comp. Meth. Appl. Mech. Engng.*, Vol. 195, No. 44, pp.6011–6045.
- Guillard, H. and Farhat, C. (2000) 'On the significance of the geometric conservation law for flow computations in moving boundaries', *Comp. Meth. Appl. Mech. Engng.*, Vol. 190, No. 11, pp.1467–1482.
- Hamamoto, M., Ohta, Y., Hara, K. and Hisada, T. (2005) 'Design of flexible wing for flapping flight by fluid-structure interaction analysis', in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation, 2005, ICRA 2005*, pp.2253–2258.
- Hatamipour, V.D., Carrington, D.B. and Heinrich, J.C. (to be published) 'Three-dimensional ALE-FEM method for fluid flow in domains with moving boundaries part II: accuracy and convergence.
- Heinrich, J.C. and Pepper, D.W. (1999) *Intermediate Finite Element Methods: Heat Transfer and Fluid Flow Applications*, Taylor and Francis, Philadelphia.
- Hirt, C.W., Amsden, A.A. and Cook, J.L. (1974) 'An arbitrary Lagrangian-Eulerian computing method for all flow speeds', *J. Comp. Phys.*, Vol. 14, No. 2, pp.227–253.
- Hua, C., Fang, C. and Cheng, J. (2011) 'Simulation of fluid-solid interaction on water ditching of an airplane by ALE method', *Journal of Hydrodynamics, Ser. B*, October, Vol. 23, No. 5, pp.637–642.
- Johnson, A. and Tezduyar, T. (1994) 'Mesh strategies in parallel finite element computations of flow problems with moving boundaries and interfaces', *Comp. Methods Appl. Mech. Engng.*, Vol. 119, No. 1, pp.73–94.
- Juric, D. and Tryggvasson, G. (1996) 'A front tracking method for dendritic solidification', *J. Comp. Phys.*, Vol. 123, No. 1, pp.127–148.
- Kock, S.A., Nygaard, J.V., Eldrup, N., Fründ, E.-T., Klærke, A., Paaske, W.P., Falk, E. and Kim, W.Y. (2008) 'Mechanical stresses in carotid plaques using MRI-based fluid-structure interaction models', *Journal of Biomechanics*, Vol. 41, No. 8, pp.1651–1658.
- Löhner, R., Cebral, J.R., Camelli, F.F., Baum, J.D. and Mestreau, E.L. (2007) 'Adaptive imbedded/immersed unstructured grid techniques', *Arch. Comput. Methods Eng.*, Vol. 14, No. 3, pp.279–301.
- Mittal, R. and Iaccarino, G. (2005) 'Immersed boundary methods', *Annual Review of Fluid Mechanics*, Vol. 37, pp.239–261.
- Quartapelle, L. (1993) *Numerical Solution of the Incompressible Navier-Stokes Equations*, Basel, Birkhauser.
- Rakopoulos, C.D. and Mavropoulos, G.C. (1996) 'Study of the steady and transient temperature field and heat flow in the combustion chamber components of a medium speed diesel engine using finite element analyses', *Int. J. Energy Res.*, Vol. 20, No. 5, pp.437–464.
- Riley, N. and Drazin, P. (2006) *The Navier-Stokes Equations. A Classification of Flows and Exact Solutions*, Cambridge University Press.
- Saksono, P.H., Dettmer, W.G. and Perić, D. (2007) 'An adaptive remeshing strategy for flows with moving boundaries and fluid-structure interaction', *Int. J. Num. Meth. Eng.*, Vol. 71, No. 9, pp.1009–1050.
- Shyy, W., Udaykumar, H.S., Rao, M.M. and Smith, R.W. (1996) *Computational Fluid Dynamics with Moving Boundaries*, Taylor and Francis, Philadelphia.
- Steger, J.L. and Benek, J.A. (1987) 'On the use of composite grid schemes in computational aerodynamics', *Comp. Methods Appl. Mech. Engng.*, Vol. 64, No. 1, pp.301–320.
- Strang, G. and Fix, G.J. (1973) *An Analysis of the Finite Element Method*, Prentice-Hall, Englewood Cliffs, New Jersey.
- Taira, K. and Colonius, T. (2009) 'Three-dimensional flows around low-aspect ratio flat-plate wings at low Reynolds numbers', *Journal of Fluid Mechanics*, Vol. 623, pp.187–207.
- Taylor, C.A., Hughes, T.J.R. and Zarins, C.K. (1998) 'Finite element modeling of blood flow in arteries', *Computer Methods in Applied Mechanics and Engineering*, May, Vol. 158, Nos. 1–2, pp.155–196.
- Tezduyar, T.E. (2001) 'Finite element methods for flow problems with moving boundaries and interfaces', *Archives of Computational Methods in Engineering*, Vol. 8, pp.83–130.
- The CUBIT Geometry and Mesh Generation Toolkit (2014) Sandia National Laboratories [online] <https://cubit.sandia.gov/index.html>.
- Thomas, P. and Lombard, C. (1979) 'Geometric conservation law and its application to flow computations on moving grids', *AIAA Journal*, Vol. 17, No. 10, pp.1030–1037.
- Torres, D.J. and Trujillo, M.F. (2006) 'KIVA-4: an unstructured ALE code for compressible gas flow and sprays', *J. Comp. Phys.*, Vol. 219, No. 2, pp.943–975.
- Trescher, D. (2008) 'Development of an efficient 3-D CFD software to simulate and visualize the scavenging of a two-stroke engine', *Arch. Comput. Methods Eng.*, Vol. 15, No. 1, pp.67–111.
- van Loon, R. and Sherwin, S.J. (2006) 'A fluid-structure interaction model of the aortic heart valve', *Journal of Biomechanics*, Vol. 39, Supplement 1, p.S293.
- Wang, C.-Y. (1976) 'The squeezing of a fluid between two plates', *ASME J. Appl. Mech.*, Vol. 43, No. 4, pp.579–582.
- Yang, S.L., Peschke, B.D. and Hanjalic, K. (2000) 'Second-moment closure model for IC engine flow simulations using KIVA code', *Jour. Eng. Gas Turbines & Power, ASME*, Vol. 122, No. 2, pp.355–363.
- Zhao, P. and Heinrich, J.C. (2001) 'Front-tracking finite element method for dendritic solidification', *J. Comp. Phys.*, Vol. 173, No. 2, pp.765–796.

AN ADAPTIVE FINITE ELEMENT METHOD WITH DYNAMIC LES FOR TURBULENT REACTIVE FLOWS

Jiajia Waters,^{1,*} David B. Carrington,¹ & Darrell W. Pepper²

¹Los Alamos National Laboratory, T-3 Solid Mechanics and Fluid Dynamics, Los Alamos, New Mexico, USA

²Nevada Center for Advanced Computational Methods, Department of Mechanical Engineering, UNLV, Las Vegas, Nevada, USA

*Address all correspondence to: Jiajia Waters, E-mail: wangj16@unlv.nevada.edu

A Vreman dynamic subgrid scale (SGS) large eddy simulation (LES) model is implemented in a predictor-corrector split h-adaptive finite element method (FEM) for modeling combustion. The use of h-adaptation provides a measurement of the actual error in the discretization, and can adjust spatial accuracy to control the error. By utilizing the dynamic model, laminar or turbulent flow can be automatically calculated. In this study, we try to validate this Vreman SGS LES model in just the fluid dynamics system by assuming all species are air and all results are compared with experimental data and RANS $k - \omega$ model. The model is tested by solving an 18° ramp at a Mach number of 2.25 as well as unsteady turbulent flow over a backward-facing step. The high Mach number ramp demonstrates the ability of the model to capture shocks and shock-wave/boundary layer interactions. In the high Reynolds number backward-facing step, large eddies are resolved without the requirement of a fine mesh, in contrast to DNS, although it is finer than a RANS model would require. The reattachment length and instantaneous flow results compare well with published simulations and experimental data.

KEY WORDS: dynamic LES, adaptive finite element, turbulence, reactive flow, predictor-corrector split

1. INTRODUCTION

The combustion of fuel in an engine involves turbulent flows and many complicating factors which include highly nonlinear chemical kinetics, small-scale velocity and scalar-mixing, turbulence–chemistry interactions, compressibility effects, and variable inertia effects. Coupling between these processes occurs over a wide range of time and length scales. Other complications arise when multiple phases are present due to the introduction of dynamically evolving interface boundaries and the complex exchange processes that occur as a consequence. A dynamic Vreman large eddy simulation (LES) has been developed in a finite element form. In this work we investigate its ability to model turbulent flows on benchmark cases generally related to those flows found in engines and to be modeled in our combustion and engine modeling code, KIVA-hpFE (Carrington et al., 2013). Simulation results match previous efforts using RANS $k - \omega$ closure (Carrington et al., 2013) and experimental data (Vogel and Eaton, 1985; Vallet, 2008).

In the calculation of turbulent flows, the Reynolds time-averaged Navier-Stokes equations (RANS) are widely used and can yield good results for mean velocity and pressure fields when appropriate turbulence models (e.g., $k - \omega$) are employed to represent the averaged effects of turbulence. However, RANS methods are unable to capture detailed flow behaviors and particularly the unsteady turbulent structures. Combustion is notoriously difficult to model at the Reynolds-averaged level, but, the very fine mesh needed by direct numerical simulations (DNS), as well as the large number of species equations required for a realistic combustion model, currently makes this approach computationally too expensive for engineering use. LES is rapidly becoming more widely used to study combustion (e.g., Desjardins

NOMENCLATURE

\sim	designates a Favre-averaged variable	Q_j	subtest-scale heat flux vector
$-$	designates a grid-filtered variable	q_i	heat flux vector
c	sound speed (m/s)	Re	Reynolds number
C_p	specific heat capacity at constant P (J/kg·K)	\tilde{S}_{ij}	strain rate tensor (N/m ² , kg/m·s ²)
C_{vm}	Vreman fixed SGS eddy viscosity coefficient	Sc	Schmidt number
$C_{DV MG}$	Vreman dynamic SGS eddy viscosity coefficient	Sct	Subgrid-scale turbulent Schmidt number
D_j	diffusion coefficient of the j th species (m ² /s)	T	temperature (K)
D_k	turbulent diffusion coefficient (m ² /s)	T_{ij}	subgrid test-scale stress tensor
E	total internal energy (J/kg)	t_{ij}	grid scale (resolved scale) shear stress (N/m ² , kg/m·s ²)
$f_{k,j}$	body forces (N/m ³)	u_i	velocity component (m/s)
f_{drop}	body forces related to particulate or droplets in flow (N/m ³)	$\Upsilon_j f_j$	body force term for the j th component
H_j	enthalpy of species j (J)	\dot{w}_{chem}^j	chemical reaction
H_{oj}	enthalpy of formation (J)	\dot{w}_{spray}^j	spray evaporation
P	pressure (Pa)	Greek Symbols	
Pr	molecular Prandtl number	∂t	discrete time step size (s)
Pr _{sgs}	SGS eddy Prandtl number	κ	coefficient of thermal conductivity (W/m·K)
Pr _{DV MG}	Vreman dynamic SGS eddy Prandtl number	ρ	density (kg/m ³)
		Υ^j	mass fraction (j th species) (ρ^j / ρ)
		τ_{ij}	subgrid-scale stress tensor
		μ	fluid viscosity (Pa·s)
		μ_{sgs}	turbulent eddy viscosity

and Frankel, 1999; Colin et al., 2000; Pierce and Moin, 2002; Selle et al., 2004). By assuming all species are air without any chemical reactions, this paper is just a first step to show better turbulence modeling for KIVA combustion, which is part of a larger effort to enhance combustion predictability and efficiencies within engines. The temperatures and species (air) presented in this paper are not sufficient to create chemical reactions, although we carry the full chemistry and species transport equations as part of the validation process.

LES can be used whether or not the flow is compressible or heat transfer is present. Extensions of subgrid-scale models to variable-density flows are straightforward. True compressibility effects of the weak subgrid-scale motions are likely to be negligible. This assumption is valid because the growth in the compressible shear layer is not related to dissipation, but rather to production and pressure-strain. Production is determined by the large scales and initiates the turbulent energy cascade, while dissipation mainly happens at the small scales. It could indicate that the large-scale turbulence is considerably altered by compressibility while the small eddies are more incompressible; see Vreman (1995) for further explanation.

LES uses filtered equations in time and space and the method requires a finer grid than RANS, but the grid scale is not as fine as needed in DNS. The use of h -adaptation (Carrington et al., 2013) refines the mesh where the local relative error is large (measured by a percentage of the average total error in the domain). This refinement process assists LES modeling by producing a solution with a specified error on the domain utilizing a minimal number of elements, thereby reducing the computational time; i.e., it minimizes the computer time of solution for a given error in the solution (Carrington, et al., 2010). In addition, this h -adaptive method is especially helpful to capture shocks and other flow features that might not be resolvable with the grid resolution used at the start of a simulation. Results of LES applications to flows that include shock-induced boundary layer separation are also discussed.

Filtration of the conservation equations over finite mesh sizes gives rise to physical scales that are smaller than the mesh size and cannot be resolved by LES methods, subsequently requiring the use of subgrid correlations, which model those subgrid scales. Since the filtered grid scale in general needs to contain perhaps 80% of the turbulent kinetic energy, the nonresolved small scales are more isotropic than large scales (Kolmogorov, 1941; Smyth and Moum, 2000); some of the effects can be reasonably accounted for by means of a subgrid-scale (SGS) model. Modeling of the subgrid correlations is performed on the assumption that SGS can be obtained based on the information of large or resolved scales. One of the early approaches is due to Smagorinsky (1963) who developed a subgrid model, called by its acronym SM and which is still widely used. In the Smagorinsky SGS model, the eddy viscosity ν_t is modeled as $\nu_t = (C_s \Delta)^2 \sqrt{2S_{ij}S_{ij}}$, where C_s is the eddy viscosity coefficient and is assumed to be a constant, Δ is the grid size, and S_{ij} is the strain rate tensor. There are two key reasons for the success of the Smagorinsky model. First, it yields sufficient diffusion and dissipation to stabilize the numerical computations (Pope, 2000; Deardorff, 1970); second, low-order statistics of the larger eddies are usually insensitive to the subgrid-scale motions (Meneveau, 1994; Ghosal and Rogers, 1997). Coefficient C_s is also adjusted to give the best result for each flow (Rogallo and Moin, 1984). The subgrid-scale stresses vanishing in laminar flow and at a solid boundary, and having the correct asymptotic behavior in the near-wall region of a turbulent boundary layer, allow one to conclude that C_s should vary in time (Germano et al., 1991). However, most investigators choose to keep C_s constant throughout the flow or make modifications to the SGS model (Smagorinsky, 1963; Piomelli et al., 1988; Moin and Kim, 1982; Yakhot et al., 1986; Vreman, 2004).

Vreman (2004) proposed a different invariant coefficient model, here termed VM, which appears to have many advantages over SM. The Vreman model not only guarantees vanishing SGS dissipation for various laminar shear flows, but also eliminates the need to use a wall-damping function in boundary layer flows (Vreman, 2004; Lau et al., 2012). This makes the method especially suitable for LES simulations of wall-bounded shear flows. VM has also been more successful than SM in modeling highly anisotropic transitional flows and appears well suited for complex flows containing laminar, transitional, and turbulent flows (Vreman, 2004; Kemenov et al., 2012).

Simulations of unsteady incompressible and turbulent flow are presented. Several dynamic variants based on this model have also been developed which allow the model coefficient to be varied dynamically (You and Moin, 2007; 2009; Park et al., 2006). In this work, we illustrate the ability of dynamic VM-LES by Lau et al. (2012) with local adaptive refinement to solve both compressible and incompressible flows. Unlike most turbulence models, this VM-LES model does not involve any explicit filtering, averaging, or clipping procedure to stabilize the numerical procedure, enabling it to be used in simulations of reacting flows with complex geometries.

2. THE MODEL EQUATIONS

2.1 Momentum Equation

The conservation of momentum per unit volume for a fluid can be expressed as

$$\frac{\partial \rho u_i}{\partial t} = - \frac{\partial \rho u_i u_j}{\partial x_j} - \frac{\partial p}{\partial x_i} + \frac{\partial t_{ij}}{\partial x_i} + f_{drop} + \rho \sum_{k=1}^{NumSpecies} \Upsilon_k f_{k,j}, \quad (1)$$

in which the shear stress t_{ij} is

$$t_{ij} = 2\mu \frac{\partial u_i}{\partial x_j} + \lambda \frac{\partial u_k}{\partial x_k} \delta_{ij}. \quad (2)$$

f_{drop} is the momentum of spray droplets being imparted to the aggregate fluid. In this study, f_{drop} is not used. Utilizing the Stokes hypothesis with $\lambda = 2/3\mu$ when the species is a gas, yields

$$t_{ij} = \mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3} \frac{\partial u_k}{\partial x_k} \delta_{ij} \right), \quad (3)$$

for laminar flow.

2.2 Energy Conservation

The equation for total energy in conservative form including the components heat of formation of a species and body forces for species for completeness (the terms shown are present in our model for combustion modeling, although not all are investigated in this paper) is as follows:

$$\begin{aligned} \frac{\partial E}{\partial t} = & -\frac{\partial}{\partial x_i} (Eu_i + pu_i) + \frac{\partial}{\partial x_i} \kappa \frac{\partial T}{\partial x_i} + \frac{\partial t_{ij}}{\partial x_i} u_j + \frac{\partial}{\partial x_i} \left(\rho \sum_{k=1}^{NumSpecies} \bar{H}_k D_k \frac{\partial \Upsilon_k}{\partial x_i} \right) \\ & + \rho \sum_{j=1}^{NumSpecies} \Upsilon_j f_j(x_i) \cdot u_i - \sum_{k=1}^{NumSpecies} H_{o,k} w_k. \end{aligned} \quad (4)$$

2.3 Species Transport

We model species transport represented by mass fraction in the solution process. In the test cases we use air as the two constituents of the fluid. The inlet boundaries are specified as air, with the rest of the domain as a separate species of air. This is similar to specifying the fluid using molecular constituents. The mass fraction of each species is given by

$$\Upsilon_j = \frac{\rho^j}{\rho},$$

where ρ is the total fluid density. The transport equation for species is expressed by

$$\frac{\partial \rho \Upsilon_j}{\partial t} = -\frac{\partial}{\partial x_i} (\rho u_i \Upsilon_j) + \frac{\partial}{\partial x_i} \rho D_{jN} \frac{\partial \Upsilon_j}{\partial x_i} + \Upsilon_j f_j(x_i) + w'' \quad (5)$$

where D_j is the mass diffusion coefficient in Carrington et al. (2013). The source terms \dot{w}_{chem}^j and \dot{w}_{spray}^j represent chemical reactions and spray evaporation, respectively, but are not considered in this paper, and $\Upsilon_j f_j(x_i)$ is the body force term for the j th component. In this study, we have no chemical reactions, nor spray evaporation.

3. VREMAN SUBGRID-SCALE MODEL

In LES, turbulent motions of length scales smaller than the filter size Δ are removed by applying a spatial filter to the conservation equations of mass, momentum, and energy. This filtering operation is defined as

$$\bar{\varphi}(x_j, t) = \int_D G(x_j x'_j) \varphi(x'_j, t) dx'_j, \quad (6)$$

in which D is the flow domain, G is the filter function, and $x_j = (xyz)$ are Cartesian coordinates for point j . In this study, the box or top-hat filter is written in the j -direction as

$$G(x_j - x'_j) = \begin{cases} \frac{1}{\Delta_j} & \text{for } |x_j - x'_j| < \frac{\Delta_j}{2} \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

where Δ is the grid filter width (representing a length scale). Usually, in three-dimensional flows $\Delta = (\Delta x \Delta y \Delta z)^{1/3}$, but in the finite element context Δ may be taken as the cubic root of the element volume.

It is desirable to avoid having to model the subgrid scales of the continuity equation. Therefore the filtered equations are best formulated by using Favre averaging as

$$\tilde{\varphi}(x_j, t) = \frac{\rho \bar{\varphi}(x_j, t)}{\bar{\rho}},$$

in which $\tilde{\varphi}(x_j, t)$ is a Favre-filtered variable obtained from applying averaging on its grid-filtered component $\bar{\varphi}(x_j, t)$.

3.1 Turbulent Flow with Multi-Species

The Favre-filtered continuity and momentum which govern the evolution of large-scale eddies are expressed as

$$\frac{\partial \bar{\rho}}{\partial t} + \frac{\partial(\bar{\rho} \tilde{u}_i)}{\partial x_i} = 0 \quad (8)$$

$$\frac{\partial(\bar{\rho} \tilde{u}_i)}{\partial t} + \frac{\partial(\bar{\rho} \tilde{u}_i \tilde{u}_j)}{\partial x_j} = \frac{\partial \tilde{t}_{ji}}{\partial x_j} - \frac{\partial \bar{p}}{\partial x_i} + \frac{\partial \tau_{ji}}{\partial x_j} + \bar{f}_{drop} + \bar{\rho} \sum_{k=1}^{NumSpecies} \tilde{\Upsilon}_k f_{k,j} \quad (9)$$

where the body forces related to droplet or particulate are shown for completeness of model equations in the code, although not invoked in this paper. The stress tensor, \tilde{t}_{ij} , is evaluated using the Stokes hypothesis as

$$\tilde{t}_{ij} = \mu \left(\frac{\partial \tilde{u}_i}{\partial x_j} + \frac{\partial \tilde{u}_j}{\partial x_i} \right) - \frac{2}{3} \mu \frac{\partial \tilde{u}_k}{\partial x_k} \delta_{ij} \quad (10)$$

The energy equations which govern the evolution of large-scale eddies are expressed as

$$\begin{aligned} \frac{\partial \tilde{E}}{\partial t} = & -\frac{\partial}{\partial x_i} \left(\tilde{E} \tilde{u}_i + p \tilde{u}_i \right) + \frac{\partial}{\partial x_i} \kappa \frac{\partial \tilde{T}}{\partial x_i} - \frac{\partial (C_p q_i)}{\partial x_i} + \frac{\partial}{\partial x_i} (t_{ij} + \tau_{ij}) \\ & + \frac{\partial}{\partial x_i} \left[\bar{\rho} \sum_{j=1}^{NumSpecies} \bar{H}_K \left(D_K + \frac{\mu_{sgs}}{Sc_t} \right) \frac{\partial \tilde{\Upsilon}_K}{\partial x_i} \right] + \bar{\rho} \sum_{j=1}^{NumSpecies} \tilde{\Upsilon}_j f_j(x_i) \cdot \tilde{u}_i - \sum_{k=1}^{NumSpecies} H_{o,k} w_k \end{aligned} \quad (11)$$

As mentioned, the fluid in the domain is represented as two species, both air, each placed in different regions of the domain, inlet, and the remainder of the domain. Although we simply could have chosen one species, our studies presented in this paper help in our validation of the region setup processes and partially investigate the aggregation of species (if the aggregation process does not return the material properties for air when both species are air, an error surely exists in the material property evaluation algorithm or method).

The turbulent species equation has the same form as the thermal energy transport equation given by

$$\frac{\partial \bar{\rho} \tilde{\Upsilon}_j}{\partial t} = -\frac{\partial}{\partial x_i} \left(\bar{\rho} \tilde{u}_i \tilde{\Upsilon}_j \right) + \frac{\partial}{\partial x_i} \bar{\rho} \left[\left(D_{j,N} + \frac{\mu_{sgs}}{Sc_t} \right) \frac{\partial \tilde{\Upsilon}_j}{\partial x_i} \right] + \bar{\rho} \tilde{\Upsilon}_j f_j(x_i) + \dot{w}_{chem}^j + \dot{w}_{spray}^j \quad (12)$$

In Eqs. (8)–(12), $\bar{\rho}$ is the filtered density, \tilde{T} is the filtered temperature, \tilde{u}_i is the filtered velocity vector, \tilde{E} is the filtered energy, $\tilde{\Upsilon}$ is the filtered species, Pr is the molecular Prandtl number, μ is the dynamic viscosity, C_p is the specific heat capacity at constant pressure, and Sc_t is the turbulent Schmidt number. The SGS stress tensor τ_{ij} and SGS heat flux vector q_i in Eqs. (9) and (11) are defined, respectively, as

$$\tau_{ij} - \frac{1}{3} \tau_{kk} \delta_{ij} = -2\mu_{sgs} \left(\tilde{S}_{ij} - \frac{1}{3} \tilde{S}_{kk} \delta_{ij} \right) \quad (13)$$

$$q_j = -\frac{\mu_{sgs}}{Pr_{sgs}} \frac{\partial \tilde{T}}{\partial x_j} \quad (14)$$

where μ_{sgs} is the turbulent eddy viscosity, Pr_{sgs} is the SGS Prandtl number, and

$$\tilde{S}_{ij} = \frac{1}{2} \left(\frac{\partial \tilde{u}_i}{\partial x_j} + \frac{\partial \tilde{u}_j}{\partial x_i} \right) \quad (15)$$

is the strain rate tensor.

3.2 Vreman SGS Model with a Fixed Model Coefficient

In the fixed model coefficient SGS developed by Vreman (2004), the SGS viscosity is determined as

$$\mu_{\text{sgs}} = \bar{\rho} C_{vm} \Pi^g \quad (16)$$

in which

$$\Pi^g = \sqrt{\frac{B_\beta^g}{\alpha_{ij}\alpha_{ij}}} \quad (17)$$

$$\alpha_{ij} = \frac{\partial \tilde{u}_j}{\partial x_i} \quad (18)$$

$$\beta_{ij} = \sum_{m=1}^3 \Delta_m^2 \alpha_{mi} \alpha_{mj}, \quad (19)$$

$$B_\beta^g = \beta_{11}\beta_{22} - \beta_{12}^2 + \beta_{11}\beta_{33} - \beta_{13}^2 + \beta_{22}\beta_{33} - \beta_{23}^2 \quad (20)$$

In this model, the superscript g denotes filtering at the grid-scale level and C_{vm} is the model coefficient. Vreman (2004) obtained good results with the Smagorinsky model for various turbulent flows using $C_{vm} = 0.07$. For the solution of turbulent energy transport the SGS Prandtl number, $\text{Pr}_{\text{sgs}} = 0.4$ was adopted as by Lau et al. (2012).

3.3 Vreman SGS Model with Dynamic Coefficient

Although the Vreman SGS model with fixed coefficient produces better results than the ad hoc modification of the Smagorinsky coefficient in transitional and turbulent flows, Germano et al. (1991) concluded that it is impossible to find a single, universal constant for different flows. In addition, none of any fixed coefficient SGS models can account for the energy transfer from unresolved to resolved scales (backscatter), which may also occur intermittently, while on average, energy is transferred from the large to the small scales (forward scatter) (Piomelli et al., 1991). The development of the dynamic subgrid-scale Smagorinsky model (DSGS) reflects significant progress in the subgrid-scale modeling of nonequilibrium flows. The DSGS model can be used to calculate the eddy viscosity coefficient by sampling the smallest resolved scale, rather than by setting a priori parameters. The second filter is larger than the grid filter with the grid filter equal to the resolved scale. The idea is to minimize the difference between this larger test filter and grid filter. The dynamic SGS stress model uses this minimization process to model the local subgrid eddy viscosity and form the proper coefficient. This is performed by sampling the smallest resolved scales and using this information to create the model for the subgrid scales, providing closure to the turbulence model.

The DSGS is obtained by two filtering processes: in the first process, the grid filter Δ is applied, where the filtered expressions are given by (8)–(12), with the SGS Reynolds stress included. In the second process, a test filter $\hat{\Delta} = 2\Delta$ is added to the grid-filtered equations (8)–(12), leading to the subtest-scale stress tensor \hat{T}_{ij} and subtest-scale heat flux vector \hat{Q}_j :

$$T_{ij} - \frac{1}{3} T_{kk} \delta_{ij} = -2\mu_{\text{sgs}} \left(\hat{S}_{ij} - \frac{1}{3} \hat{S}_{kk} \delta_{ij} \right) \quad (21)$$

and

$$Q_j = -\frac{\mu_{\text{sgs}}}{\text{Pr}_{\text{sgs}}} \frac{\partial \hat{T}}{\partial x_j} \quad (22)$$

Here we define $\mu_{\text{sgs}} = \bar{\rho} C_{\text{DVMG}} \Pi^t$ and $\text{Pr}_{\text{sgs}} = \text{Pr}_{\text{DVMG}}$, and $\hat{S}_{ij} = \frac{1}{2} \left(\frac{\partial \hat{u}_i}{\partial x_j} + \frac{\partial \hat{u}_j}{\partial x_i} \right)$ is the test-filtered strain rate tensor. Using the Germano et al. (1991) identity and the least-squares error minimization technique of Lilly (1992), the coefficients C_{DVMG} and Pr_{DVMG} are obtained (see Lau et al., 2012) as

$$C_{\text{DVMG}} = \frac{\langle L_{ij} M_{ij} \rangle_V}{\langle M_{ij} M_{ij} \rangle_V} \quad (23)$$

and

$$\text{Pr}_{\text{DVMG}} = \frac{\langle M_j^\theta M_j^\theta \rangle_V}{\langle L_j^\theta M_j^\theta \rangle_V} \quad (24)$$

respectively, where

$$M_{ij} = -2 \left(\hat{\rho} \Pi^t \hat{S}_{ij} - \widehat{\rho \Pi^g \tilde{S}_{ij}} \right) \quad (25)$$

$$M_j^\theta = -C_{\text{DVMG}} \left(\hat{\rho} \Pi^t \frac{\partial \hat{T}}{\partial x_j} - \widehat{\rho \Pi^g \frac{\partial \tilde{T}}{\partial x_j}} \right) \quad (26)$$

$$\Pi^t = \sqrt{\frac{B_\beta^t}{\hat{\alpha}_{ij} \hat{\alpha}_{ij}}} \quad (27)$$

$$B_\beta^t = \beta_{11}^t \beta_{22}^t - \beta_{12}^t \beta_{12}^t + \beta_{11}^t \beta_{33}^t - \beta_{13}^t \beta_{13}^t + \beta_{22}^t \beta_{33}^t - \beta_{23}^t \beta_{23}^t \quad (28)$$

$$\beta_{ij}^t = \sum_{m=1}^3 \hat{\Delta}_m^2 \hat{\alpha}_{mi} \hat{\alpha}_{mj} \quad (29)$$

$$\hat{\alpha}_{ij} = \frac{\partial \hat{u}_j}{\partial x_i} \quad (30)$$

$$L_{ij} = \widehat{\tilde{\rho} \tilde{u}_i \tilde{u}_j} - \frac{1}{\tilde{\rho}} \widehat{\tilde{\rho} \tilde{u}_i \tilde{\rho} \tilde{u}_j} \quad (31)$$

$$L_j^\theta = \widehat{\tilde{\rho} \tilde{u}_j \tilde{T}} - \frac{1}{\tilde{\rho}} \widehat{\tilde{\rho} \tilde{u}_j \tilde{\rho} \tilde{T}} \quad (32)$$

where $\langle \cdot \rangle_V$ is the volume integral over the entire domain to mitigate the effect of locally (highly) oscillating eddy viscosity fields. The DSGS coefficient remains the same throughout the entire domain and only varies in time.

4. BENCHMARK RESULTS

In order to compare results with previous work, a similar set of problems is used to illustrate the superiority of the dynamic LES Vreman model versus the RANS $k - \omega$ model (Carrington et al., 2013). Solutions are presented using the adaptive FEM with VM-LES for two flow regimes: (1) 2D incompressible flow with turbulence to demonstrate the ability of LES to capture smaller eddies in the instantaneous flow while the RANS $k - \omega$ model will only show a static recirculation throughout the entire flow process; (2) 2D supersonic compressible flow to show the shock–boundary layer interaction which will not be seen in the RANS $k - \omega$ model. We also illustrate how local h –adaptation refines the mesh in both problems to help capture eddies and the shock–boundary layer, starting with a relatively coarse mesh.

4.1 2D Flow over a 2D Backward-Facing Step

This problem is truly three-dimensional in nature. However, due to a lack of sufficient experimental data, we assume a plane of symmetry. Three-dimensional flow features typically persist for long periods of time and are unsteady. But along the centerline, the recirculation length and temperatures downstream of the expansion have been measured, providing for an average of the fluctuating values.

The first simulation is carried out to compare with experimental data obtained by Vogel and Eaton (1985), including calculations using the RANS $k - \omega$ model discussed in Carrington et al. (2013). The Reynolds number is 28,000, determined by the step height, with 296 K fluid entering the domain; $\text{Pr} = 0.71$. The bottom wall downstream from the step has a heat flux of 273 W/m² applied with all other walls being insulated. The step height (H) is 0.025 m, and the step is at $x = 0.3$ m. At the outflow boundary (located 30 step heights downstream from the expansion) a normal

gradient of zero is applied for both the velocity and energy transport equations. The flow is solved with two species, both air, to partially test the species aggregation process, where the inlet boundary is specified with one species and the rest of the domain with another species, although both species are air (see Carrington et al., 2013, for more details). LES yields 2D solutions of the velocity and pressure at each grid point and for each time step. In order to compare the result with experimental data and our previous published results by the RANS $k - \omega$ model, mean quantities were obtained by averaging these values (e.g., velocity and temperature) in time. The averaging process is starting from the beginning of the simulation and for each time step, there is a time-averaged value, e.g.: for any time t_n (n th time step),

$$\widehat{U_n} = \frac{U_0 \times \Delta t_0 + U_1 \times \Delta t_1 + \dots + U_n \times \Delta t_n}{\Delta t_0 + \Delta t_1 + \dots + \Delta t_n} \quad (33)$$

where $\widehat{U_n}$ is the time-averaged value (e.g., velocity or temperature) at time step n , U_i is the quantity value at time step i , and Δt_i is the time step size for time step i . Equation (33) is done for each point on this domain. When the time-averaged quantities are not changing, we assume the flow is stable, although the instantaneous flow may not be steady.

Throughout the solution process use of two levels of grid refinement was sufficient to accurately resolve the formation of eddies in the simulation as shown in Fig. 2. Figure 1(a) shows two levels of mesh refinement around the area containing eddies, where you can see the mesh is denser in such areas as behind the step and at the top, to the right. The mean streamlines in Fig. 1(b) show that reattachment stops at $x = 0.5$ m, and the step is at $x = 0.3$ m. Reattachment length x_R/H is $(0.5 - 0.3) / 0.025$ m, which is about 8 for the dynamic LES Vreman model which agrees with another LES model by Kobayashi (1992) who states that almost all LES models tend to overpredict the reattachment length x_R while the RANS $k - \omega$ model in Carrington et al. (2013) gives 7.2 and the experiment gives a value of about 7. However, the VM-LES model is closer to the experiment than other LES models (see Kobayashi, 1992). Also, the Smagorinsky model produces different reattachment lengths for different constants (e.g., $x_R/H = 9$ for $c = 0.1$ and $x_R/H = 8$ for $c = 0.15$). The VM-LES appears to dynamically select the optimal constant for the simulation comparing to other LES models. Because the flow behind the step is unsteady, the dynamic coefficient varies with time and does not converge to a steady value. Backscatter phenomena are also depicted with the DSGS coefficient being negative; thus the eddy viscosity is negative.

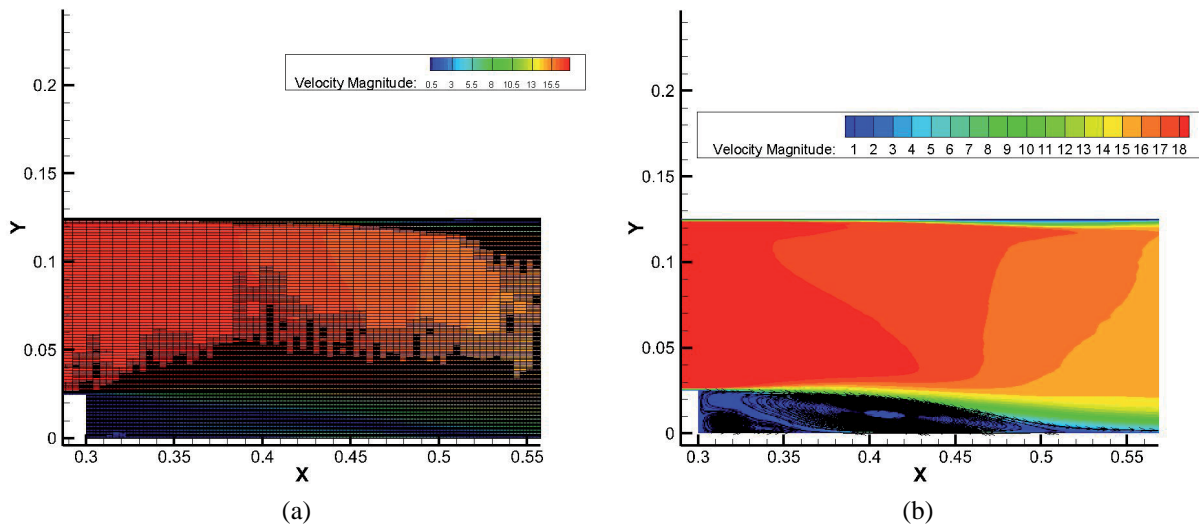


FIG. 1: 2D turbulent backward-facing step at $Re = 28,000$ using dynamic VM-LES model and h -adaptive. (a) Adapted grid; (b) mean streamline distributions

Instantaneous streamlines, obtained with the dynamic LES-VM SGS model are shown in Fig. 2 for different simulation times of $t = 0.5$ s, $t = 0.6$ s, and $t = 0.8$ s. There is a large-scale eddy in the recirculating region which contains smaller eddies, with an additional eddy downstream of the expansion step. Once an eddy in the recirculation zone downstream of the expansion reaches the reattachment length, the eddy breaks away and is convected downstream, eventually dissipating, as seen in Fig. 2(a). The location of reattachment varies slightly due to the size of the eddies in the recirculation zone due to the unsteady nature of the flow. When the eddies are shed away around at $x = 0.5$ m, the reattachment length range is $x_R/H = 8$ as shown in Figs. 2(b) and 2(c) at time $t = 0.6$ s and $t = 0.8$ s.

Figure 3(a) shows the instantaneous isotachs along with how eddies are formed behind the step (in a continual manner) at near $x = 0.3$ m to $x = 0.5$ m. Once an eddy reaches $x = 0.5$ m, it separates from the recirculation regions containing eddies, dissipating as they travel downstream to the outflow. Comparing to the $k - \omega$ two-equation model based on Reynolds time averaging (RANS), the velocity from the implemented LES model as shown in Fig. 3(b)

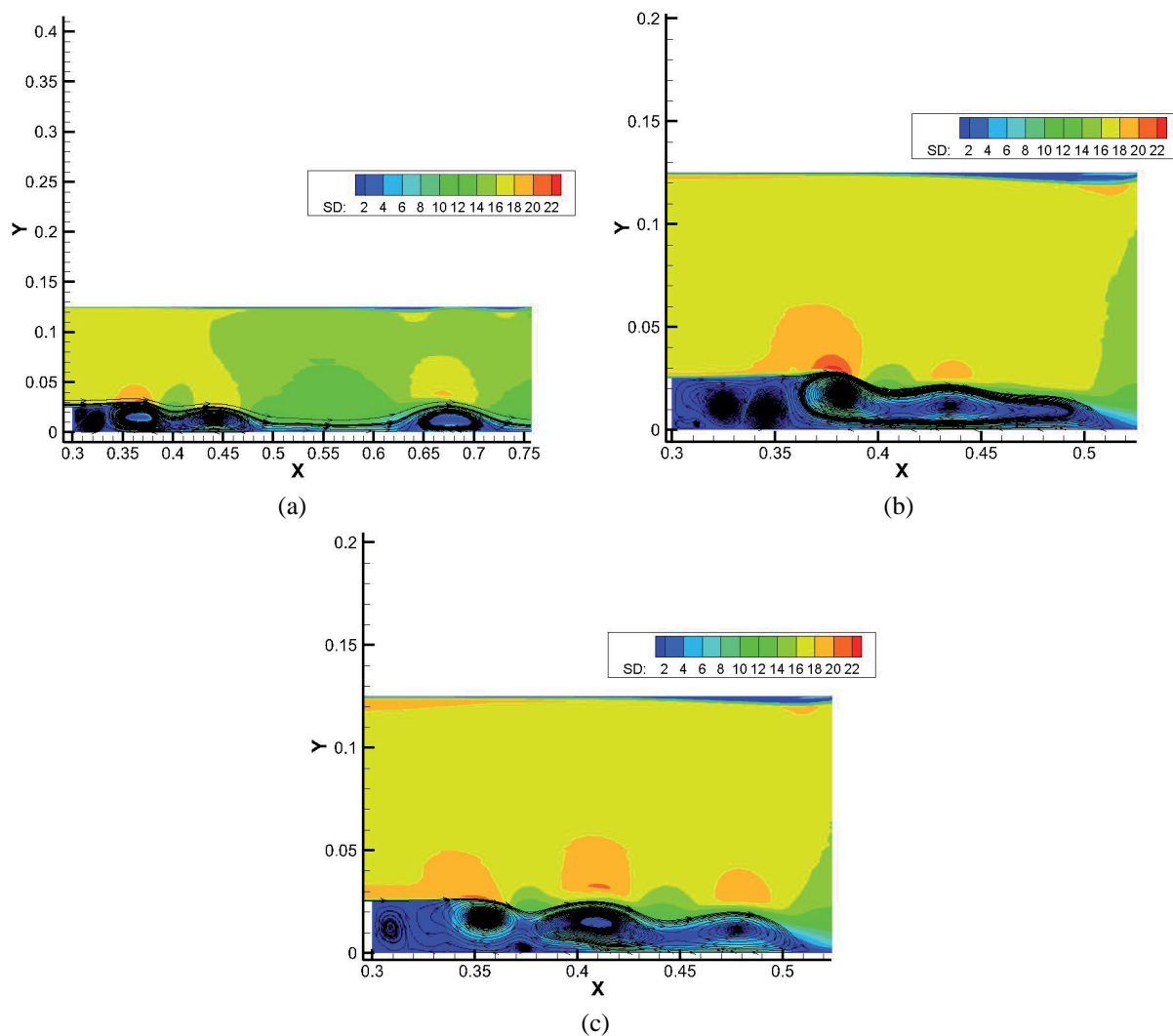


FIG. 2: Instantaneous streamlines at different times: (a) $t = 0.5$ s, (b) $t = 0.6$ s, (c) $t = 0.8$ s, instantaneous values of the streamline for the backward-facing step at $Re = 28,000$ using a dynamic VM-LES and h -adaptive PCS FEM. SD is the speed magnitude

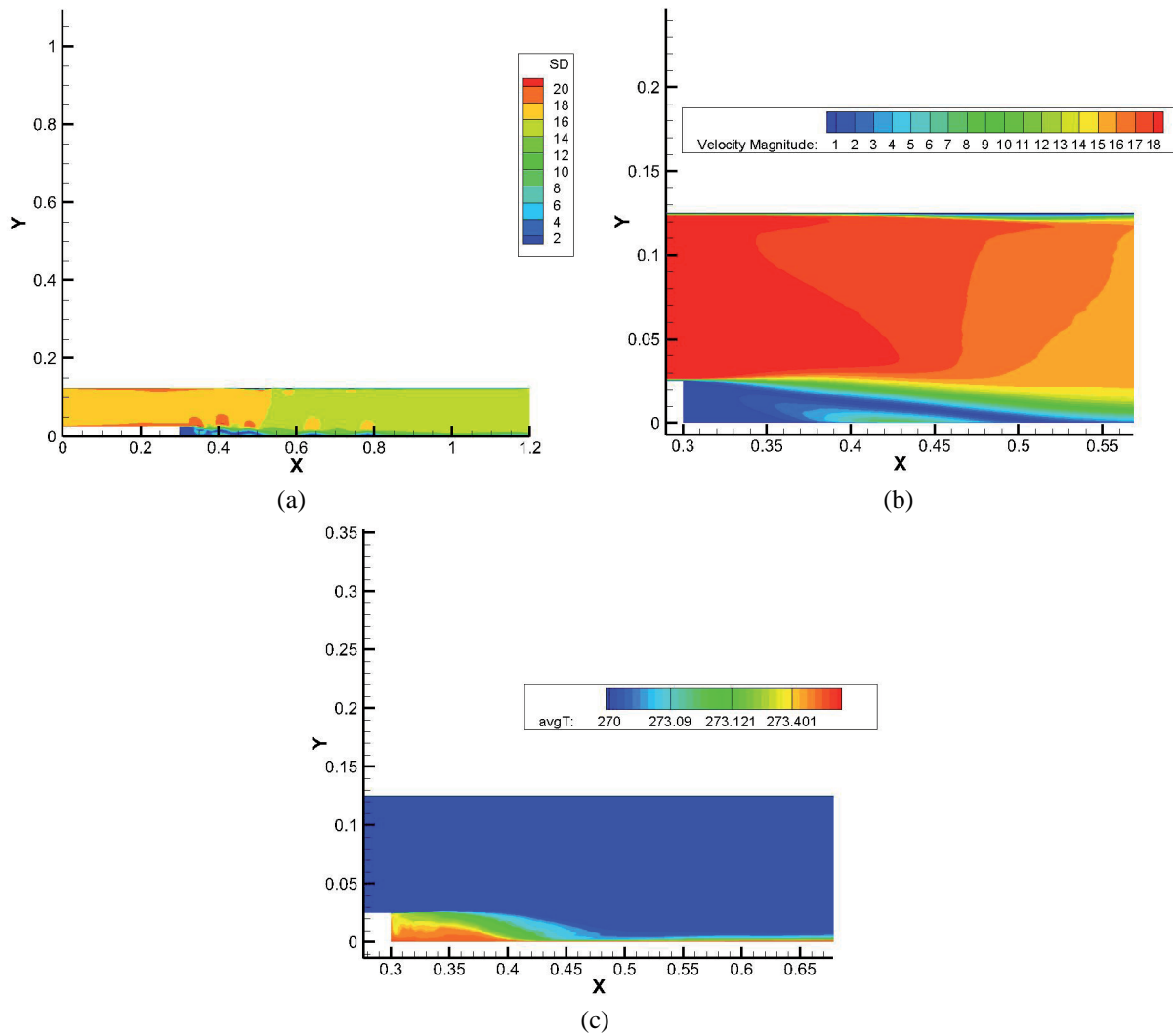


FIG. 3: (a) Velocity contour plot at $t = 1.2$ s, (b) mean velocity plot, (c) mean temperature for the backward-facing step at $Re = 28,000$ using a dynamic VM-LES and h -adaptive PCS FEM

compares favorably with results shown by Carrington et al. (2013) and experimental data as shown in Fig. 4. This comparison is performed by the averaging process described in Eq. (33) over sufficient duration to include eddy developments and separations. The mean velocity reaches its steady state around $t = 0.4$ s and the reattachment stays the same as Fig. 3(b) until the end of the simulation elapsed time of 1.3 s. Averaging temperature over the same period of elapsed time produces the isotherms shown in Fig. 3(c).

In Fig. 4, the mean velocity and mean temperature profiles at different locations, respectively, are compared with measured values obtained by Vogel and Eaton (1985). In Fig. 4(a), $x^* = (x - x_r)/x_r$, where x_r is the reattachment point, in our case $x_r = 0.5$ m, e.g., $x^* = 0.33$ is $0.33 = (x - 0.5)/0.5$, then $x = 0.665$ m. The vertical axis in Fig. 4 is the actual y in the simulation domain divided by the step height $H = 0.025$ m in our simulation; e.g., $Y/H = 1$ is $Y = 0.025$ m. The same notations are used in Fig. 4(b). The difference between our numerical results and experimental data is felt to be attributed to introduced by the extrapolation of experimental data from 3D to 2D. Dotted lines are experimental data from Vogel and Eaton (1985) and solid lines are from the numerical model.

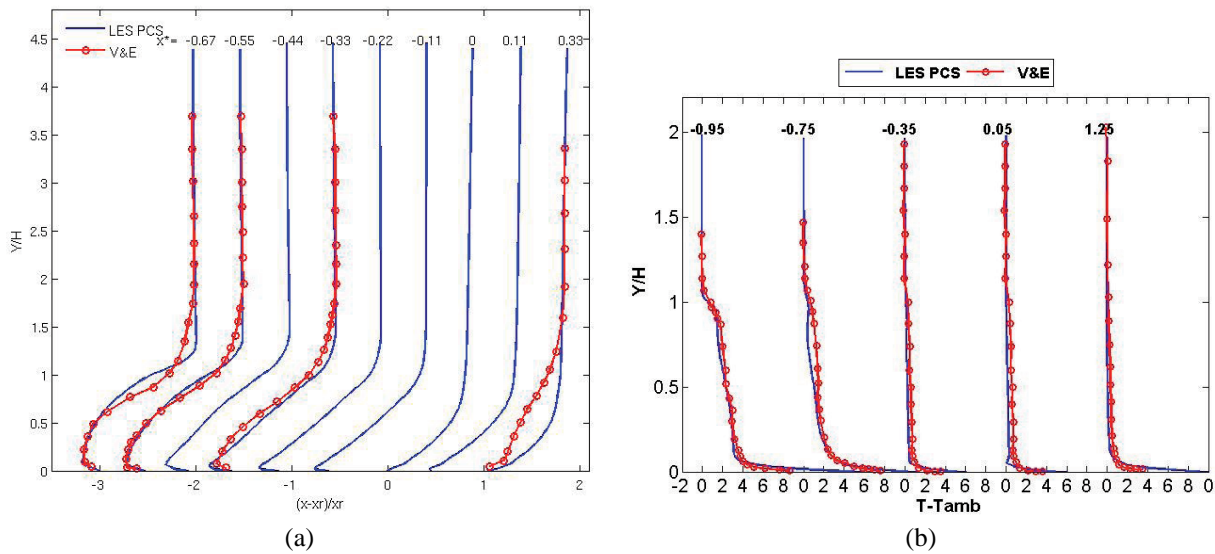


FIG. 4: (a) Mean U-component of velocity profile, **(b)** mean temperature profiles for the backward-facing step at $Re = 28,000$ using a dynamic VM-LES and h -adaptive PCS FEM compared to the experimental data

The results indicate that the dynamic LES-VM-SGS model is more accurate than other LES and the RANS $k - \omega$ models in demonstrating instantaneous flow for unsteady incompressible flow. At this point, we now test the compressible flow model.

4.2 Supersonic Compression Ramp at Mach 2.25

The second example problem deals with viscous supersonic flow over a compression ramp. The ideal gas law is used to evaluate density once the pressure is determined in the PCS solver, and then the temperature is extracted from specific internal energy assuming a calorically perfect gas. A ramp at 18° is simulated and the result is compared with the data by Vallet (2008). As the compressible flow over the ramp reaches steady state, the dynamic eddy viscosity coefficient C_{DVMG} varies between 0.11 and 0.13 and Pr_{DVMG} varies between 0.3 and 0.32 when the flow becomes fully developed.

Figure 5 shows various contours with streamlines plotted for the recirculation at the ramp corner. Upstream of the shock and downstream from the compression wave (between the two), a flow reversal occurs very near the bottom wall caused by the existence of an adverse pressure gradient formed in the boundary layer (shown in Fig. 5). The point where the recirculation begins upstream of the expansion step is in agreement with experimental data occurring at 0.021 m upstream of the ramp, as shown in Fig. 5(a). There is a second eddy between the tip of the shock and the boundary layer shown in Fig. 5(b), which illustrates the interaction between the shock and the boundary layer. This second eddy cannot be captured by the RANS $k - \omega$ model in Carrington et al. (2013). Other turbulence models utilizing wall-damping techniques typically have the assumption of zero shear stress near the boundary, omitting turbulence generation in the boundary layer area region above the laminar sublayer. Figure 6 shows additional plots for velocity x components U , density, and local Mach number, and all appear to be in relatively good agreement with results from RANS $k - \omega$ turbulence models and experimental data.

The mean velocity values in the boundary layer are shown in Fig. 7 for the PCS dynamic VM-LES and results appear comparable and follow similar trends as the experimental data. The differences among the results from our dynamic LES-VM-SGS model and the experimental data are also seen between the RANS $k - \omega$ model and the experimental data described in Carrington et al. (2013). Some of these differences likely attributed to the method used to extract the data from the experiment and the errors associated with the experimental procedures. However, while

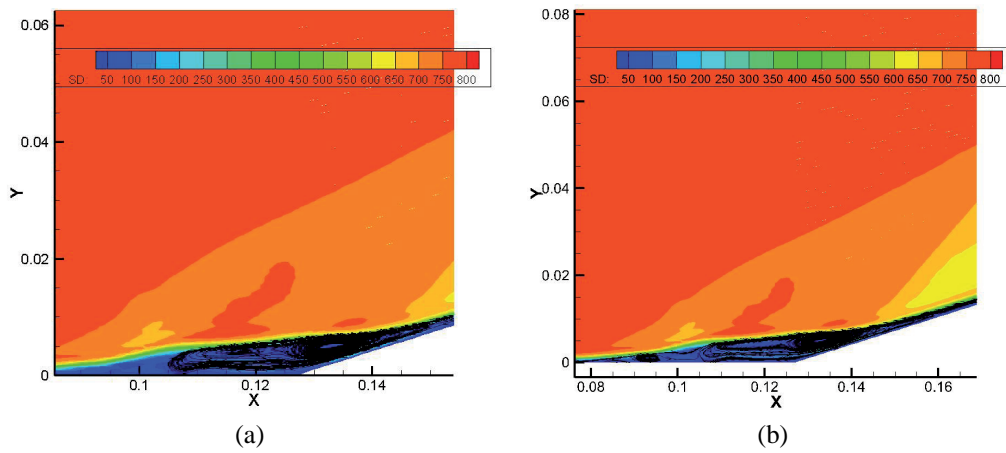


FIG. 5: Detail of flow reversal between the shock wave and wall for Mach 2.25 over a 18° ramp. **(a)** Recirculation without showing the second eddy; **(b)** recirculation showing the second eddy at the tip of the shock between 0.09 and 0.1 m

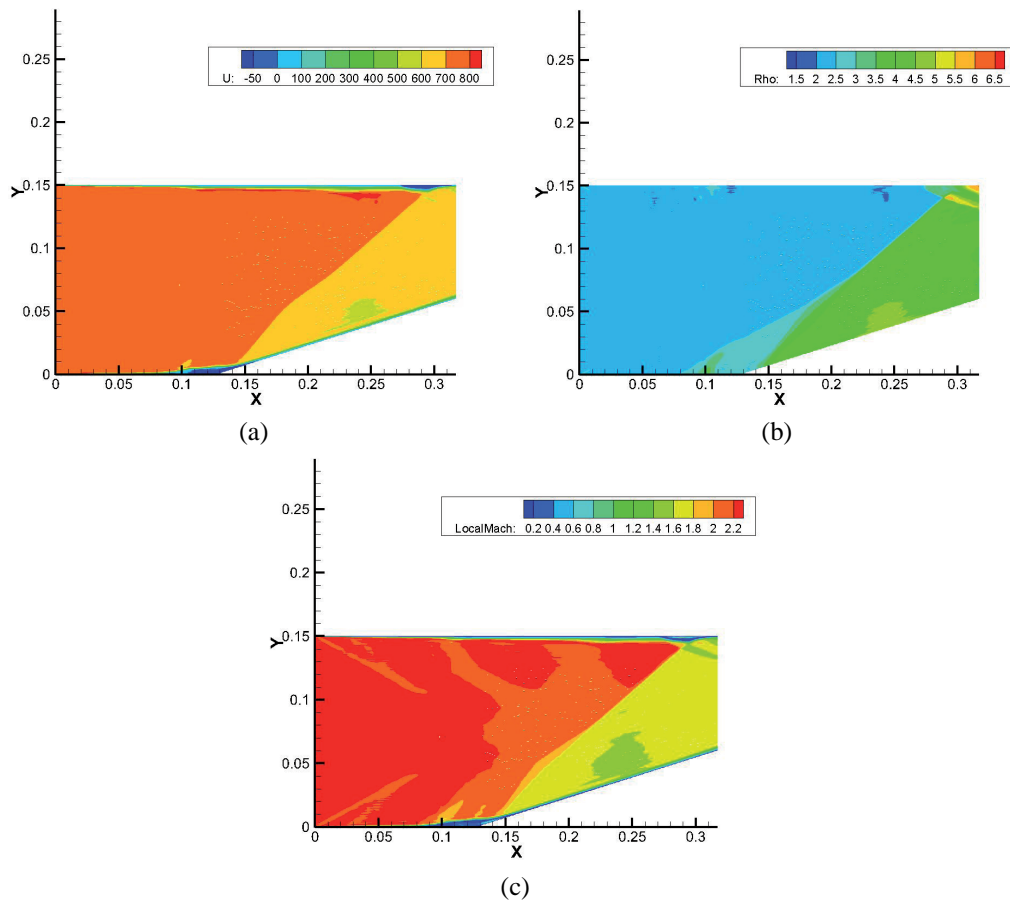


FIG. 6: Mach 2.25 steady-state flow properties for supersonic viscous flow through an 18° compressible ramp: **(a)** U-component of the velocity; **(b)** density; **(c)** local Mach number

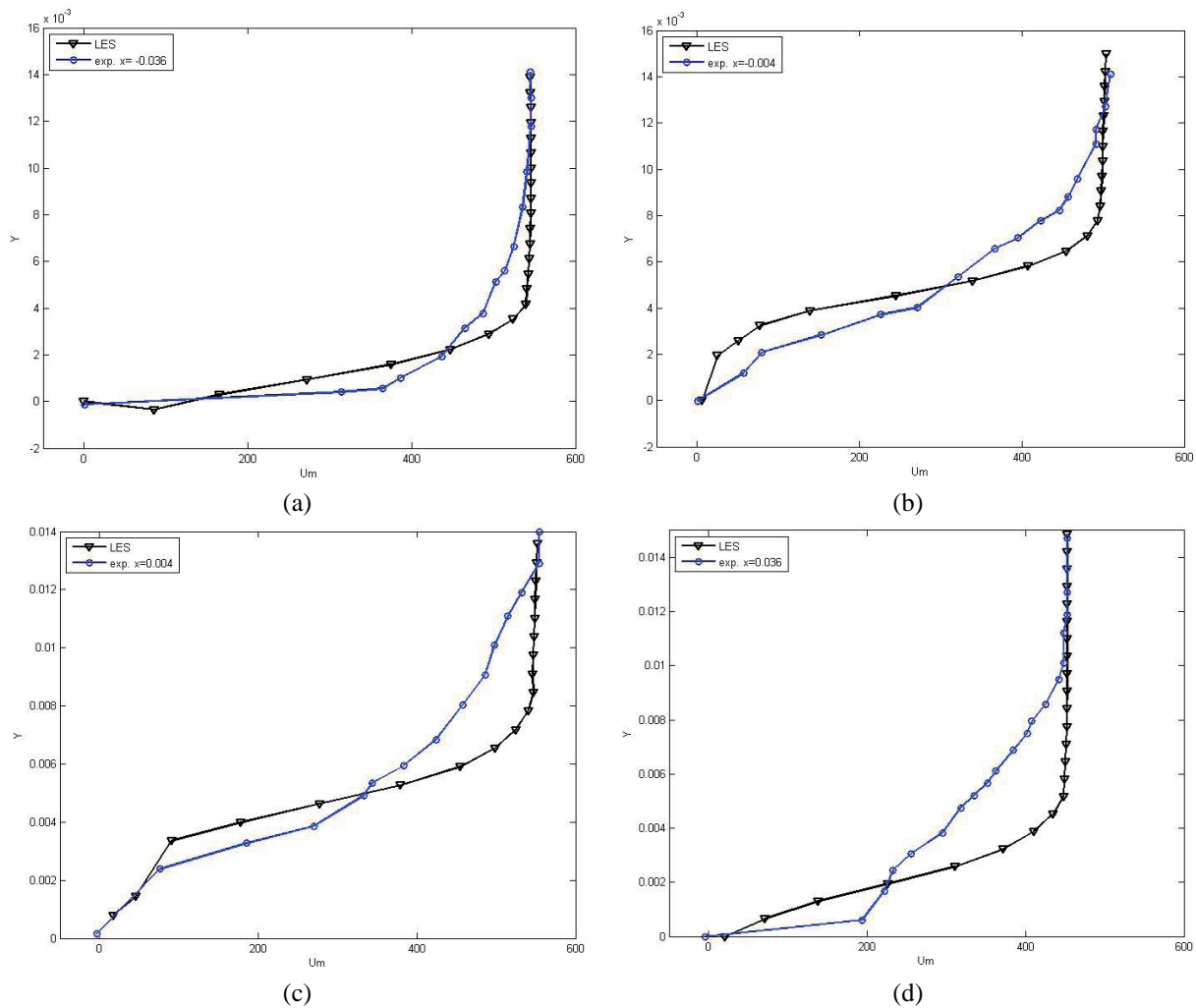


FIG. 7: Mean velocity U in the bottom boundary layer using the dynamic VM-LES model; comparison with data by Vallet (2008) at various location upstream (–) and downstream (+) of the ramp: (a) -0.032 m; (b) -0.004 m; (c) $+0.004$ m; (d) $+0.032$ m

we obtain the same accuracy as the RANS $k - \omega$ model, more detail is shown in the shock–boundary layer interaction within the second eddy. The use of adaptive elements significantly aids in capturing this interaction. The VM-LES model permits more detail to be captured between the shock and boundary layer interaction without the need for a wall function, or a damping function, to reduce the dissipation effects in the sublayer. The dynamic VEM-LES model detects the boundary layer and the h –adaptation method locally refines the mesh, allowing more accurate interaction to be undertaken between the shock and boundary layer.

5. CONCLUSION

An adaptive PCS FEM with dynamic VM-LES method is used to simulate both incompressible and compressible flows and the solutions are compared to well-known benchmark values. This present work demonstrates the numerical implementation employing local adaptation with a refined LES technique that produces accurate results without wall-

layer calculations. Compared with the RANS $k - \omega$ closure scheme, the dynamic VM-LES is easier to implement and is able to model the unsteady nature of turbulent flow, having advantages when small variations in fluid properties, including local turbulence and vorticity, are important in the problem being modeled, e.g., species variations, and reactive flow. The local adaptive process reduces the computing burden of the LES, compared with a globally fine mesh.

Turbulence development in time is shown by solving unsteady incompressible turbulent flow associated with the backward-facing step, a well-known benchmark test problem. After taking the time average of all quantities, results were compared with experimental data. Since the flow is unsteady, the DSGS coefficient varies as the flow develops which can reflect closely the state of the flow and is also capable of accounting for backscatter. While a substantial amount of research has been carried out using LES for modeling incompressible flows, applications to compressible flows have been significantly fewer. With the PCS solver, compressible flow can be solved without introducing extra unclosed terms for the SGS stresses and SGS heat flux. The ability of the h -adaptation dynamic VM-LES model to refine the grid and thereby capture flow phenomena, such as shocks and shock-boundary layer interactions, was demonstrated. Results show good agreement with our previous work using the RANS $k - \omega$ closure and experimental data. Finer scales of turbulence are captured by the LES in the shock-boundary layer region than by the RANS $k - \omega$ model. The DSGS coefficient converges for the compression ramp problem once it reaches steady state.

LES is becoming more widely used to study combustion in many modern combustion devices. The Vreman dynamic LES model has attributes favorable for turbulence modeling. Even though multiple proofs of the validity of the LES concept have been shown for gaseous combustion, LES for combustion remains an area of interest, for which very few studies are available. The LES method appears to offer advantages in modeling unsteady flows with turbulence and offer more accurate and predictive simulations of turbulent combustion. Future work will extend this LES model to reactive flows.

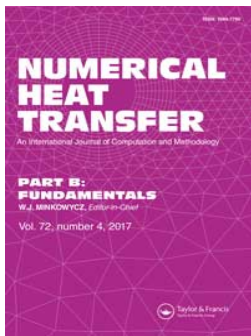
ACKNOWLEDGMENTS

We wish to thank Dr. John Reizes (SIT; UNSW) for his suggestion and comments regarding the Vreman LES model, and the DOE's Office of Energy Efficiency and Renewable Energy (EERE) Advanced Combustion Program (Gurpreet Singh and Leo Breton) for support.

REFERENCES

- Carrington, D. B., Wang, X., and Pepper, D. W., An h -adaptive finite element method for turbulent heat transfer, *Comput. Modeling Eng. Sci. (CMES)*, vol. **61**, no. 1, pp. 23–44, 2010.
- Carrington, D. B., Wang, X., and Pepper, D. W., A predictor-corrector split projection method for turbulent reactive flow, *Comput. Therm. Sci.*, vol. **5**, no. 4, pp. 333–353, 2013.
- Colin, O., Ducros, F., Veynante, D., and Poinot, T., A thickened flame model for large eddy simulations of turbulent premixed combustion, *Phys. Fluids*, vol. **12**, pp. 1843–1863, 2000.
- Deardorff, J., A numerical study of three-dimensional turbulent channel flow at large Reynolds numbers, *J. Fluid Mech.*, vol. **41**, no. 2, pp. 453–480, 1970.
- Desjardins, P. E. and Frankel, S. H., Two-dimensional large eddy simulation of soot formation in the near field of a strongly radiating non-premixed acetylene-air jet flame, *Combust. Flame*, vol. **119**, pp. 121–133, 1999.
- Germano, M., Piomelli, U., Moin, P., and Cabot, W. H., A dynamic subgrid-scale eddy viscosity model, *Phys. Fluids A*, vol. **3**, no. 7, pp. 1760–1765, 1991.
- Ghosal, S. and Rogers, M. M., A numerical study of self-similarity in a turbulent plane wake using large-eddy simulation. *Phys. Fluids*, vol. **9**, pp. 1729–1739, 1997.
- Kemenov, K. A., Wang, H., and Pope, S. B., Turbulence resolution scale dependence in large-eddy simulations of a jet flame, *Flow Turbul. Combust.*, vol. **88**, no. 4, pp. 529–561, 2012.
- Kobayashi, T., Morinishi, Y., and Oh, K.-J., Large eddy simulation of backward-facing step flow, *Commun. Appl. Numer. Methods*, vol. **8**, pp. 431–441, 1992.

- Kolmogorov, A. N., Dissipation of energy in locally isotropic turbulence, *Proc. R. Soc. London, Ser. A*, vol. **434**, pp. 15–17, 1941.
- Lau, G. E., Yeoh, G. H., Timchenko, V., and Reizes, J. A., Application of dynamic global-coefficient subgrid-scale models to turbulent natural convection in an enclosed tall cavity, *Phys. Fluids (1994-present)*, vol. **24**, 094105, 2012.
- Lilly, D. K., A proposed modification of the Germano subgrid scale closure method, *Phys. Fluids A*, vol. **4**, pp. 633–635, 1992.
- Meneveau, C., Statistics of turbulence subgrid-scale stresses: Necessary conditions and experimental tests, *Phys. Fluids*, vol. **6**, pp. 815–833, 1994.
- Moin, P. and Kim, J., Numerical investigation of turbulent channel flow, *J. Fluid Mech.*, vol. **118**, pp. 341–377, 1982.
- Park, N., Lee, S., Lee, J., and Choi, H., A dynamic subgrid-scale eddy viscosity model with a global model coefficient, *Phys. Fluids*, vol. **18**, 125109, 2006.
- Pierce, C. D. and Moin, P., Progress-variable approach for large eddy simulation of nonpremixed turbulent combustion, *J. Fluid Mech.*, vol. **504**, pp. 73–97, 2004.
- Piomelli, U., Moin, P., and Ferziger, J. H., Model consistency in large eddy simulation of turbulent channel flows, *Phys. Fluids*, vol. **31**, pp. 1884–1891, 1988.
- Piomelli, U., Cabot, W., Moin, P., and Lee, S., Subgrid-scale backscatter in turbulent and transitional flows, *Phys. Fluids A*, vol. **3**, no. 7, pp. 1766–1771, 1991.
- Rogallo, R. S. and Moin, P., Numerical simulation of turbulent flows, *Ann. Rev. Fluid Mech.*, vol. **16**, pp. 99–137, 1984.
- Pope, S., *Turbulent Flows*, Cambridge, Cambridge University Press, 2000.
- Selle, L., Lartigue, G., Poinso, T., Koch, R., Schildmacher, K.-U., Krebs, W., Prade, B., Kaufmann, P., and Veynante, D., Compressible large-eddy simulation of turbulent combustion in complex geometry on unstructured meshes, *Combust. Flame*, vol. **137**, pp. 489–505, 2004.
- Smagorinsky, J., General circulation experiments with the primitive equations. I. The Basic Experiment, *Mon. Weather Rev.*, vol. **91**, pp. 99–164, 1963.
- Smyth, W. D. and Moum, J. N., Anisotropy of turbulence in stably stratified mixing layers, *Phys. Fluids*, vol. **12**, pp. 1343–1362, 2000.
- Vallet, I., Reynolds-stress modeling of $M=2.25$ shock-wave/turbulent boundary-layer interaction, *Int. J. Numer. Methods Fluids*, vol. **56**, pp. 525–555, 2008.
- Vogel, J. C. and Eaton, J. K., Combined heat transfer and fluid dynamic measurements downstream of a backward facing step, *J. Heat Transfer*, vol. **107**, pp. 922–929, 1985.
- Vreman, A. W., Direct and large-eddy simulation of the compressible turbulent mixing layer, Ph.D. Dissertation, University of Twente, 1995.
- Vreman, A. W., An eddy-viscosity subgrid-scale model for turbulent shear flow: Algebraic theory and applications, *Phys. Fluids*, vol. **16**, pp. 3670–3681, 2004.
- Yakhot, A., Orszag, S. A., Yakhot, V., and Israeli, M., Renormalization group formulation of large-eddy simulations, *J. Sci. Comput.*, vol. **4**, pp. 139–158, 1989.
- You, D. and Moin, P., A dynamic global-coefficient subgrid-scale eddy-viscosity model for large-eddy simulation in complex geometries, *Phys. Fluids*, vol. **19**, 065110, 2007.
- You, D. and Moin, P., A dynamic global-coefficient subgrid-scale model for large eddy simulation of turbulent scalar transport in complex geometries, *Phys. Fluids*, vol. **21**, 045109, 2009.



Numerical Heat Transfer, Part B: Fundamentals

An International Journal of Computation and Methodology

ISSN: 1040-7790 (Print) 1521-0626 (Online) Journal homepage: <https://www.tandfonline.com/loi/unhb20>

Modeling multiphase flow: Spray breakup using volume of fluids in a dynamics LES FEM method

Jiajia Waters, David B. Carrington & Marianne M. Francois

To cite this article: Jiajia Waters, David B. Carrington & Marianne M. Francois (2017) Modeling multiphase flow: Spray breakup using volume of fluids in a dynamics LES FEM method, Numerical Heat Transfer, Part B: Fundamentals, 72:4, 285-299, DOI: [10.1080/10407790.2017.1400307](https://doi.org/10.1080/10407790.2017.1400307)

To link to this article: <https://doi.org/10.1080/10407790.2017.1400307>



Published online: 17 Nov 2017.



Submit your article to this journal [↗](#)



Article views: 136



View Crossmark data [↗](#)



Citing articles: 2 View citing articles [↗](#)



Modeling multiphase flow: Spray breakup using volume of fluids in a dynamics LES FEM method

Jiajia Waters^a, David B. Carrington^a, and Marianne M. Francois^b

^aLos Alamos National Laboratory, T-3 Division, Los Alamos, New Mexico, USA; ^bLos Alamos National Laboratory, XCP-4, Methods and Algorithms, Los Alamos, New Mexico, USA

ABSTRACT

Currently, all commercial software for engine modeling investigates the dispersed droplet phase of the injection process. Understanding the effect of geometry of the injector nozzle, initial jet conditions, fluid properties in the liquid film, breakup, resulting droplet sizes, and distribution are of primary importance to improve fuel efficiency and lower gas emissions. We have developed an innovative computational method and models to make this atomization process more predictive: a multiscale, multiphase fluid simulation, using a volume-of-fluid method implemented in a large eddy simulation algorithm found in the new KIVA-hpFE, a finite element method flow solver for all flow regimes.

ARTICLE HISTORY

Received 10 August 2017
Accepted 26 October 2017

1. Introduction

Sprays are of primary importance in engine combustion, particularly in modern fuel injected engine where efficiency and low emission is the goal, and of paramount importance. Improvements in engine and fuel efficiency have a grand economic and environmental impact, where a mere gallon of fuel saved per mile, decreases fuel consumption enormously, essentially providing hundreds of millions of dollars in economic stimulus in the USA alone, while decreasing greenhouse gas and no emissions into the atmosphere. Engine designers and researchers in the industry have relied on expensive and limited range experiments to determine how best to design and implement the spray injection process. Modeling of sprays has extended the operating range of the experimental results helping to provide faster design progress and changes inclusive of computer system control of injection and variable injection operation.

The earlier KIVA codes (II, 3, 3v by Amsden et al. [1]) and KIVA-4 by Torres and Trujillo [2] use Lagrangian particle transport (LPT) for the droplet-phase simulation. The particle transport portion of the LPT model was developed by Dukowicz [3]. This method is also currently used in the new KIVA-hpFE code by Carrington [4]. Solution of the dispersed spray equation requires initial conditions for the droplets after it transforms from a continuously connected fluid. Atomization of an injected liquid occurs after the liquid is forced through a nozzle, forming a liquid core and subsequent ligamentation created by the stresses that the liquid core is experiencing. This ligamentation breaks into fine droplets as a result of the stresses as well where the liquid drops begin to experience convective evaporation processes. The developed multiscale, multiphase method as described in this paper helps remove uncertainty of initial conditions from the LPT system used in the dispersed droplet modeling. Initial phase-space information for solving the LPT multicomponent spray equation will be supplied without a priori information of the breakup process and the model tuning required to meet experimental injection measurements. Geometrical concerns and fluid dynamic behavior of

Nomenclature

C_p	specific heat capacity at constant P (J/kg · K)	Greek symbols	
c	sound speed (m/s)	ρ	density (kg/m ³)
E	total internal energy (J/kg)	β	artificial compressibility
e	any element	Y_j	mass fraction
f_s	surface tension force	σ	surface tension coefficient
\hat{n}	unit normal of the interface surface	δ_Γ	Dirac delta function
P	pressure (Pa)	κ	surface curvature
T	temperature (K)	μ	fluid viscosity (PaΔs)
t_{ij}	grid-scale (resolved scale) shear stress	μ_{sgs}	turbulent eddy viscosity
u_i^*	$\left(\frac{N}{m^2} \cdot \frac{kg}{m} m\Delta s^2\right)$ intermediate velocity	ν	kinematic viscosity
V_e	volume of computational element	τ_{ij}	subgrid-scale stress tensor
		ϕ	volume of fraction
		ϕE	elemental value for volume of fraction

the initial injection process will be properly modeled, mitigating uncertainty of their effects in the spray's downstream behavior. The proposed methods will remove a need for extensive experimentation as geometries and operation conditions change from cycle to cycle and from design to design. The innovative and novel methods will result in reduced computational time and improved accuracy for realistic simulation of sprays from injector to evaporation.

The basic hydrodynamic instabilities responsible for the jet breakup are based on shear instability (Kelvin–Helmholtz) and density instability (Rayleigh–Taylor). Understanding the effect of geometry of the injector nozzle, the initial jet conditions, fluid properties (in particular, density, viscosity, and surface tension) on the liquid film breakup, and the resulting droplet sizes and distribution are of primary importance to improve fuel efficiency and lower gas emissions. For the new “clean fuels,” it is important to understand how fuel properties will affect spray breakup mechanisms and the subsequent engine and operation designs. The principal objective of this research is to model the interface dynamics with an interface tracking method until the point where the interface has broken up into multiple droplets smaller than the grid size (i.e., when the interface is no longer tractable). In this work, an interface tracking method (specifically a volume of fluid [VOF] method) is considered as a first step toward coupling it to the particle tracking method of the KIVA code by Torres et al. [5] and Carrington [4]. For the transition model, we will develop methods based on error quantification to determine whether we should switch to a Lagrangian formulation for simulating the spray.

To capture the interface dynamics, we rely on the VOF method. A velocity and pressure field formulation for the entire domain is used for solving the mass, momentum conservation equations on a fixed computational mesh. An evolving “color” field equation is solved, depicting the interface kinematics. In the VOF method, the color function is the fractional volume (amount) of each fluid in a computational cell. The VOF method is intrinsically mass conservative and automatically handles changes in topology (breakup and coalescence), since it solves an evolution equation for the volume fraction field. VOF is a multiscale method which leads to higher fidelity of the interfacial region and is fundamental to accurately model coupled physics (e.g., to determine reaction rate in chemical reaction), see Lebas et al. [6], Shinjo and Umemura [7], and Xiao et al. [8].

Modeling two distinct fluids in a single domain is not straightforward and requires new research. VOF is initially applied for solving the VOFs at the center of the cell, e.g., finite volume method (FVM) and conservation laws for mass and momentum need to be satisfied at the interface between computational cells using different formulations [e.g., Hirt et al. [9], Francois et al. [10], and Herrmann]. Therefore, an integral part of the flow solver is the estimate of truncated flux volumes for advective mass and momentum transport. The flux between cells with different formulations requires consideration of how the flux volumes are apportioned, and how the destination

cell receives mass and momentum to ensure conservation [Kim and Lee [11]]. To describe the interface motion precisely, handle jump conditions at the interface without artificial smoothing and respect mass conservation, the level set (LS) algorithms is incorporated into VOF [Ménard et al. [12]] and Vaudor et al. [13] to perform the interface tracking from the distance function while VOF methods are to ensure mass conservation. For finite element method (FEM), flux concerns are eliminated because of the mathematical form of the FEM discretization, being automatically conserved by that formulation. However, if using a standard Galerkin FEM, it is difficult to capture the sharp interface because of numerical dispersion in the advection operator, being equivalent to the cell-center differencing with special considerations. Stabilization is required to remove this dispersion. Therefore, we use the Petrov–Galerkin FEM of Yu and Heinrich [14] that was designed to mitigate dispersion in this portion of equations, the advection, so that a wave is propagated as such, with very little dispersion, sharp as possible (three cells). A dynamic Vreman large eddy simulation (LES) (Waters et al. [15]) is adopted to turbulence modeling to capture unsteady effects on the interface dynamics. Also the use of LES model will get the statistic of the flow behavior with less requirement of resolution. The solution involves conservation of mass, turbulent momentum equations, conservation of energy, species transport, and volume fraction of material phases. Locally, when the fluids are identified as “separated,” or immiscible, the volume tracking is used to compute the properties of the two fluids for mass, momentum equation. The liquid is injected into compressible gas, so the compressible flow equations are considered when it is needed, unlike other works in the literature that assume constant temperature and use the incompressible equations everywhere.

To capture small droplets for a 0.1–0.2-mm diameter nozzle at the injection speeds of 10–100 m/s or more, a grid spacing 10th of this nozzle diameter is required everywhere, which means having several hundred million grid points. To save computational time, we adopt a nonuniform grid strategy, that is, we just need to make sure the nozzle is about one-tenth gridded, and only test the numerical simulation to see whether wave creation on ligaments is seen and middle-sized (grid size) oscillating droplets can be observed. When statistics are sufficient to show, for example, a normal distribution of ligaments, those ligaments can be passed to the LPT secondary breakup model with accompanying phase-space information [Herrmann [16]], which will be our future work. In reality, there are many mechanisms that cause instabilities of the liquid jet such as inner nozzle turbulence, cavitation, and fuel supply oscillation, but in this study, induced stresses on the liquid jet interacting with the gas as the liquid jet is penetrating the gas are considered. In this paper, we present the governing equations first and then explain the surface tension and induced stress calculations followed by numerical results.

2. Governing equations and numerical method

2.1. Model equations and solution algorithm for multiphase turbulent flow

- **VOF equation:**

Since the fluid remains constant along particle paths, we only need to account for the traveling of the liquid. In that sense, the VOF is passively advected under the incompressible flow environment:

$$\frac{\partial \phi}{\partial t} + \mathbf{U} \cdot \nabla \phi = 0 \quad (1)$$

where ϕ is the volume of fraction. Integrating ϕ on each element to get the elemental value ϕE for ϕ :

$$\phi E = \frac{\int_e \phi dV}{V_e} \quad (2)$$

where e is any element and V_e is the volume of element e .

- **Momentum equations** (FEM projection method):

If $\phi = 1$, run this incompressible equation for u_i^* , which is the intermediate velocity for predictor–corrector split projection method:

$$\frac{\rho^n u_i^* - \rho^n u_i^n}{\Delta t} = -\rho^n U \cdot \nabla u_i + \left(\mu + \mu_{\text{sgs}} \right) \frac{\partial^2 u_i}{\partial^2 x_j} + f_s \quad (3)$$

If $\phi \neq 1$, run this compressible equation for u_i^* :

$$\frac{\rho^n u_i^* - \rho^n u_i^n}{\Delta t} = -\rho^n U \cdot \nabla u_i + \frac{\partial t_{ij}}{\partial x_j} + \frac{\partial \tau_{ij}}{\partial x_j} + f_s \quad (4)$$

where t_{ij} is the shear stress, $t_{ij} = \mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3} \frac{\partial u_k}{\partial x_k} \delta_{ij} \right)$ and τ_{ij} is the Reynolds stress tensor, $\tau_{ij} = \mu_{\text{sgs}} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3} \frac{\partial u_k}{\partial x_k} \delta_{ij} \right) - \frac{2}{3} \rho K \delta_{ij}$. U is the velocity vector. ρ^n is the density at time step n , and u_i^n is the velocity at time step n . f_s is the surface tension force. Density ρ and viscosity μ are defined as:

$$\rho(\phi) = \phi \times \rho_{\text{liquid}} + (1 - \phi) \times \rho_{\text{gas}} \quad (5)$$

$$\mu(\phi) = \phi \times \mu_{\text{liquid}} + (1 - \phi) \times \mu_{\text{gas}} \quad (6)$$

Note that when $\phi = 1$, density ρ and viscosity μ are all liquid, and when $\phi \neq 1$, ρ and μ will change according to (5) and (6). The separation of momentum equations is consistent of the calculation of VOF since VOF is calculated in the assumption of the incompressible flow. Also, changing density and viscosity according to the VOF when compressible flow equations are considered complies with the physics where when $\phi \neq 1$, there is gas involved and gas density should be defined by the pressure and temperature at that point and compressibility should be taken into account. However, when $\phi = 1$, it is all liquid, which is incompressible flow, the compressibility should not be considered in the momentum equation.

- **Pressure solve:**

Let $\Delta u^* = \rho^n u_i^* - \rho^n u_i^n$ and $\Delta P = P^{n+1} - P^n$. P^n is the pressure at time step n . Since $\rho^n u_i^{n+1} - \rho^n u_i^* = -\Delta t \frac{\partial P'}{\partial x_i}$, where $P' = \theta_2 P^{n+1} + (1 - \theta_2) P^n$ and $\frac{1}{c^2} \Delta P = \Delta p = -\Delta t \frac{\partial \rho^n u_i'}{\partial x_i}$, where $\rho^n u_i' = \theta_1 \rho^n u_i^{n+1} + (1 - \theta_1) \rho^n u_i^n$, then we have

$$\frac{1}{c^2} \Delta P - \Delta t^2 \theta_1 \theta_2 \frac{\partial^2 \Delta P}{\partial^2 x_i} = \Delta t^2 \theta_1 \frac{\partial^2 P^n}{\partial^2 x_i} - \Delta t \left(\theta_1 \frac{\partial \Delta u^*}{\partial x_i} + \frac{\partial \rho^n u_i^n}{\partial x_i} \right) \quad (7)$$

Therefore, $P^{n+1} = P^n$, c is the sound speed. When it is compressible flow, sound speed is calculated by $c = \sqrt{\gamma R T}$, where R is the specific gas constant and T is the temperature. When it is incompressible flow, we use artificial compressibility β for c .

$$\beta = \max(\varepsilon, u_{\text{conv}}, u_{\text{diff}}) \quad (8)$$

where ε is a small constant to ensure β is not approaching zero. $u_{\text{conv}} = |U| = \sqrt{u_i u_i}$ and $u_{\text{diff}} = \frac{\nu}{h}$, where h is the element size and ν is the kinematic viscosity. The time step for the artificial compressibility method may be limited by the artificial compressibility β , which could be big ($\beta \rightarrow \infty$). Therefore, pseudo-time-stepping is needed in the incompressible region; refer to Zienkiewicz et al. [17] for more details about artificial compressibility. To balance the pressure at region $0 < \phi(x) < 1$, we take the control volume integral $P = \frac{\int_e p dV}{V_e}$ for every point in that area to avoid the pressure being discontinuous, where e is all of the elements associated with that point and V_e is the volume of all of those elements related to that point.

- **Velocity update:**

Update the velocity u_i^{n+1} by

$$\rho^n u_i^{n+1} - \rho^n u_i^n = \Delta u^* - \Delta t \frac{\partial P'}{\partial x_i} = \Delta u^* - \Delta t \times \left(\theta_2 \frac{\partial \Delta P}{\partial x_i} + \frac{\partial P^n}{\partial x_i} \right) \quad (9)$$

- **Energy equations:**

Solving the energy equations could be difficult in VOF sense because the heat conductivity coefficient cannot just simply be approximated according to VOF especially for two immiscible fluids. Therefore, we consider the energy in three separate flow types as:

1. $\phi = 0$: Compressible
2. $0 < \phi < 1$: Both Compressible and incompressible
3. $\phi = 1$: Incompressible

The equations we are solving for internal energy are:

If $\phi > 0$, run the incompressible equations for internal energy E_{liquid} :

$$\rho_{\text{liquid}} \frac{\partial E_{\text{liquid}}}{\partial t} = -\rho_{\text{liquid}} C_{p,\text{liquid}} U \cdot \nabla T + \frac{\partial}{\partial x_i} \left(\kappa_{\text{liquid}} + \frac{C_{p,\text{liquid}} \mu_{\text{sgs}}}{Pr_{t,\text{sgs}}} \right) \frac{\partial T}{\partial x_i} \quad (10)$$

If $\phi < 1$, run the compressible equations for internal energy E_{gas} :

$$\rho_{\text{gas}} \frac{\partial E_{\text{gas}}}{\partial t} = -\rho_{\text{gas}} U \cdot \nabla E_{\text{gas}} - P \nabla \cdot U + \frac{\partial}{\partial x_i} \left(\kappa_{\text{gas}} + \frac{C_{p,\text{gas}} \mu_{\text{sgs}}}{Pr_{t,\text{sgs}}} \right) \frac{\partial T}{\partial x_i} + \frac{\partial}{\partial x_i} (t_{ij} + \tau_{ij}) u_j \quad (11)$$

Aggregate the internal energy as:

$$E = \phi \times E_{\text{liquid}} + (1 - \phi) \times E_{\text{gas}} \quad (12)$$

Note that in the interval of $\phi \in (0, 1)$, both liquid energy and gas energy are being calculated and then the final internal energy is approximated by VOFs as Eq. (12).

- **Species equations:**

Define mass fraction as $Y_j = \frac{\rho_j}{\rho_{\text{gas}}}$, where ρ_{gas} is the gas density and ρ_j is species j . Here we only consider the gas species for mass fraction. The liquid density is tracked according to VOF. Therefore, the species equations will be:

$$\rho_{\text{gas}} \frac{\partial Y_j}{\partial t} = -\rho_{\text{gas}} U \cdot \nabla Y_j + \frac{\partial}{\partial x_i} \rho_{\text{gas}} \left(D + \frac{\mu_{\text{sgs}}}{Sc_t} \right) \frac{\partial Y_j}{\partial x_i} \quad (13)$$

This equation is not solved when $\phi = 1$.

- **Aggregation for gas properties**

Use Y_j and its species properties to aggregate the gas properties for $C_{p,\text{gas}}$, $C_{v,\text{gas}}$, κ_{gas} , and μ_{gas} , the same procedure is followed by Carrington et al. [18]. If $\phi < 1$, $\rho_{\text{gas}} = \frac{p}{RT}$. Now we can aggregate the properties by VOF

$$\mu = \phi \times \mu_{\text{liquid}} + (1 - \phi) \times \mu_{\text{gas}} \quad (14)$$

$$\rho = \phi \times \rho_{\text{liquid}} + (1 - \phi) \times \rho_{\text{gas}} \quad (15)$$

$$\text{Temperature } T = \frac{E_{\text{liquid}}}{C_{v,\text{liquid}}} \text{ for } \phi > 0 \text{ and } T = \frac{E_{\text{gas}}}{C_{v,\text{gas}}} \text{ for } \phi = 0.$$

In the governing equations, μ_{sgs} and $Pr_{t,\text{sgs}}$ are calculated the same way as Waters and Carrington [19], which is a dynamic Vreman LES model. All equations are solved under the FEM system. Specifics on the discretized FEM method are presented in many of our papers, notably Carrington et al. [18]. The VOF equation is simply the advection of wave. To get an accurate result for the VOF solution, we apply the Petrov–Galerkin in the system as described by Yu and Heinrich [14].

This scheme provides for a sharp interface no larger than three cells and having negligibly small under shot and essentially no overshoot (Gibbs effect). If it is necessary, truncation is applied to ensure that the VOF is kept in the interval $[0,1]$.

2.2. Surface tension calculation

The surface tension force is given by

$$f_s = \sigma \kappa \hat{n} \delta_\Gamma \quad (16)$$

where σ is the surface tension coefficient, κ is the surface curvature, \hat{n} is the unit normal of the interface surface, and δ_Γ is a Dirac delta function. Due to the intrinsic property of FEM, the VOF ϕ is smooth. The interface unit vector can be easily calculated as:

$$\hat{n} = \frac{\nabla \phi}{|\nabla \phi|} \quad (17)$$

at points where the VOF ϕ is between 0 and 1, and the curvature is given as:

$$\kappa = -(\nabla \cdot \hat{n}) \quad (18)$$

The surface tension is a continuum surface force (CSF) as described by Brackbill [20], and the interface Dirac delta function δ_Γ is set to be

$$\delta_\Gamma = |\nabla \phi|. \quad (19)$$

Therefore, the surface tension is viewed as a force and can be written as $\sigma \kappa \nabla \phi$. The calculations of interface unit vector \hat{n} , the curvature κ , and $\nabla \phi$ follow the discretization of FEM.

From Eq. (19), we do not need to keep track where the interface is $\nabla \phi$ will be 0 when ϕ is all 0 (all gas) or 1 (all liquid) and δ_Γ will be nonzero only in the transition region. The advantage of Petrov-Galerkin FEM is allowing ϕ not to change abruptly across the interface; hence no large inaccuracy will be introduced by the numerical discretization for the spatial derivatives of ϕ . When the discretization of ϕ is discontinuous as in a finite volume or finite difference formulations, the value of ϕ must be smoothed using some technique, which can be a delicate yet arduous process, such as using kernel or balanced force methods (Francois et al. [10], Beliveau [21], and Errmann [22]).

3. Numerical results

3.1. Surface tension calculation verification

We start with a standard test case of a 3d static drop. Since the gravity is not accounted, there is no external force and we assume it is an inviscid fluid, only the surface tension force is balanced by the pressure force. The exact pressure jump ΔP_{exact} across the interface is given by:

$$\Delta P_{\text{exact}} = \sigma \kappa$$

and the exact curvature κ in 3D is $\kappa_{\text{exact}} = 2/R$.

To have a good comparison with the other literature, a test case is used, that is the same as the case in Francois et al.'s [10] publication. The computational domain is a cube having side lengths of 8 units. The grid is $40 \times 40 \times 40$. The domain is decomposed to run in parallel on 10 processors. The drop is centered inside the cube with a radius $R = 2$. The surface tension coefficient is chosen to be $\sigma = 73$. The density inside the drop is $\rho_1 = 1$ and the background fluid density ρ_2 ranges from 1 to 0. The initial velocity and pressure are set to zero. The domain setup can be found in Figure 1.

Ideally, the pressure field should be constant inside the drop and should be exactly balanced by the surface tension force at the interface. Any velocity field is numerical artifacts. Therefore, a good way

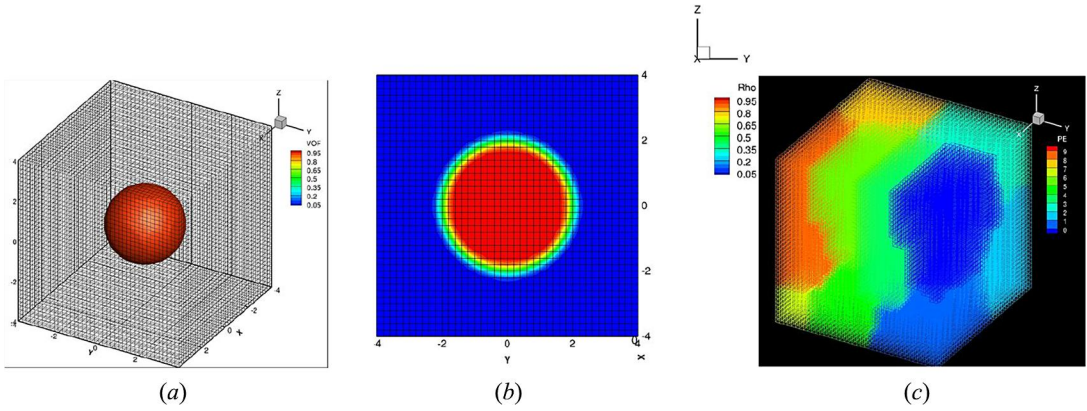


Figure 1. Initial setup and domain. (a) Static drop shown as VOF is placed in the center of the cube, (b) y-z plane cutting the drop shows density (Rho), and (c) mesh decomposition (PE is the processor number). Note: volume of fluid.

to validate the surface tension calculation algorithm is a measure of the maximal velocity in the computational field and the pressure jump given by:

$$\Delta P_{\max} = P_{\max} - P_{\min}$$

where P_{\max} is the maximum pressure and P_{\min} the minimum pressure on the domain.

3.1.1. Exact curvature

To start the validation process, the exact curvature is used to test the surface tension calculation. The results for different density ratios at different run time with exact curvature are shown in Table 1.

In Table 1, $E(\Delta P_{\max}) = \frac{|\Delta P_{\max} - \Delta P_{\text{exact}}|}{\Delta P_{\text{exact}}}$ and the steady state is when the maximal velocity $|U|_{\max}$ is not changing and ΔP_{\max} remains the same according to time. We see that from Table 1, the pressure jump is better established as the simulation runs longer. This is because initial state of the fluid volume ϕ was set to be discontinuous across the interface. The VOF ϕ value is smoothed as during the simulation with the FEM scheme. Figure 2 shows the pressure at different density ratios and we notice that inside the drop, the pressure is constant which matches the theory and only pressure jumps are seen at the interface.

The velocity vector field for density ratio 10^3 is depicted in Figure 3a, while the VOF is shown as the contour identifying the drop's location. The velocity (Figure 3) is mainly on the surface of the drop corresponding to a constant pressure within the drop. Figure 3b shows the surface tension force vector field and the magnitude of the surface tension force by isopleths. We notice that surface tension forces are symmetric along the drop and are concentrated on the interface of the drop and the surrounding media, balancing the pressure resulting in discontinuous pressure or a pressure jump.

3.1.2. Curvature estimates algorithm

Figure 4 shows the estimated curvature [from Eq. (18)] in the line contour plot overlaying the VOF calculation shown. The VOF interface ranges over three cells. The curvature calculation at the center of VOF range has a value 1, the correct estimate.

Table 1. Error in velocity and pressure for the inviscid 3-D static drop test Equilibrium when the exact curvature is specified.

ρ_1/ρ_2	After one time step		Steady state	
	$ U _{\max}$	$E(\Delta P_{\max})$	$ U _{\max}$	$E(\Delta P_{\max})$
10^3	4.6273×10^{-4}	6.2×10^{-2}	1.2386×10^{-3}	2.1918×10^{-4}
10^5	5.3053×10^{-2}	4.72×10^{-2}	1.2178×10^{-1}	3.411×10^{-4}

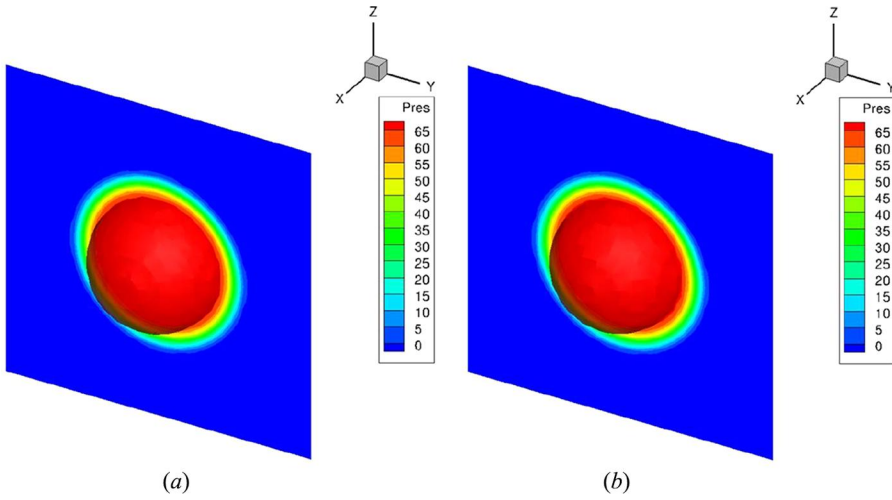


Figure 2. Pressure plots for (a) density ratio $\frac{\rho_1}{\rho_2} = 10^3$ and (b) density ratio $\frac{\rho_1}{\rho_2} = 10^5$.

Comparing in Table 2 the maximal velocity when the density ratio is 10 with balanced-force algorithm by Francois [10]. The method described in this paper produces smaller spurious velocities than those from balanced-force algorithm after first time step. The spurious velocities grow slowly after 50 time steps as compared with a height function (a better method than convolution).

Also the resolution study has been done with different grid sizes. On the cube having side lengths of 8 units, three sets of grid sizes $20 \times 20 \times 20$, $40 \times 40 \times 40$, and $80 \times 80 \times 80$ are compared as shown in Figure 5 in terms of L_∞ error in maximum velocity $|U|_{\max}$ and in the pressure jump at 50 time steps. The error is spatially convergent trending toward zero for the highest resolution.

3.2. Primary breakup modeling

Now that the validation of the VOF algorithm is established, we use the method to spray modeling and begin validating the results by comparison to direct numerical simulation (DNS). In engines and burners, fuel injection and subsequent droplet atomization to form a spray is started by injecting a liquid stream into a chamber. The liquid stream experiences pressure gradients, surface tension,

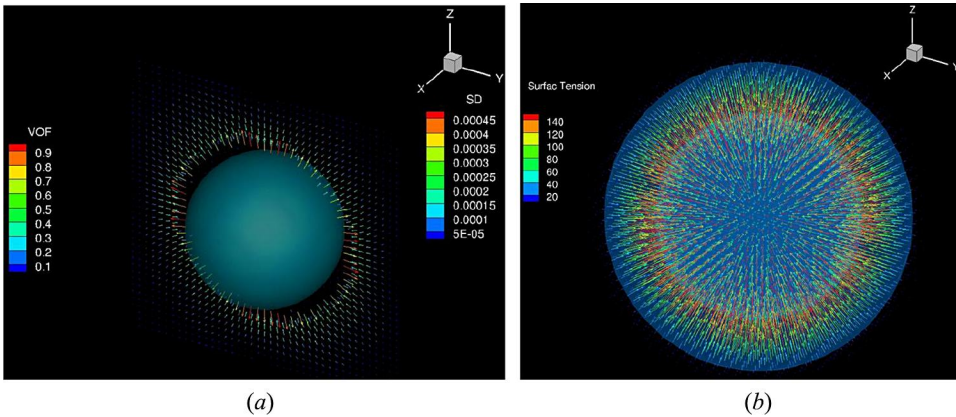


Figure 3. Density ratio 10^3 at steady-state (a) velocity vector field (SD is the magnitude of the velocity) and VOF contour plot (b) surface tension force vector field. *Note:* volume of fluid.

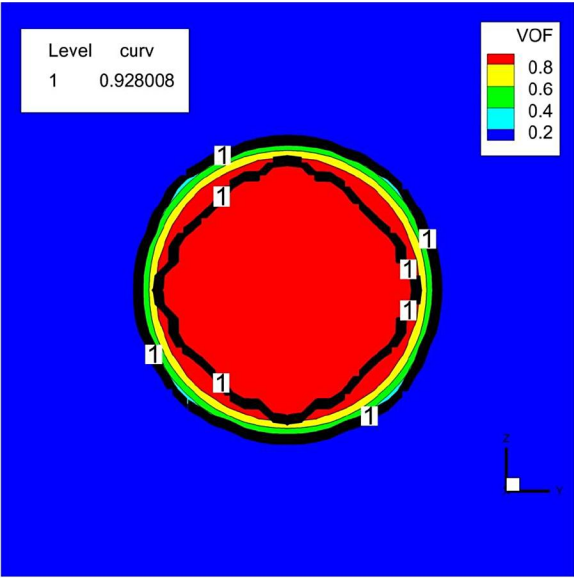


Figure 4. Curvature estimate (line contour plot) overlaying VOF identifies the drop. *Note:* volume of fluid.

Table 2. Comparing error in maximum velocity $|U|_{\max}$ after one and 50 time steps for a 3-D inviscid static drop in equilibrium using the FEM curvature estimates with other methods.

$\Delta t = 10^{-5}$	FEM algorithm	Balanced-force algorithm	
		Convolution	Height function
$t = \Delta t$	1.0684×10^{-5}	4.87×10^{-3}	4.02×10^{-3}
$t = 50\Delta t$	1.393×10^{-4}	1.63×10^{-1}	4.02×10^{-2}

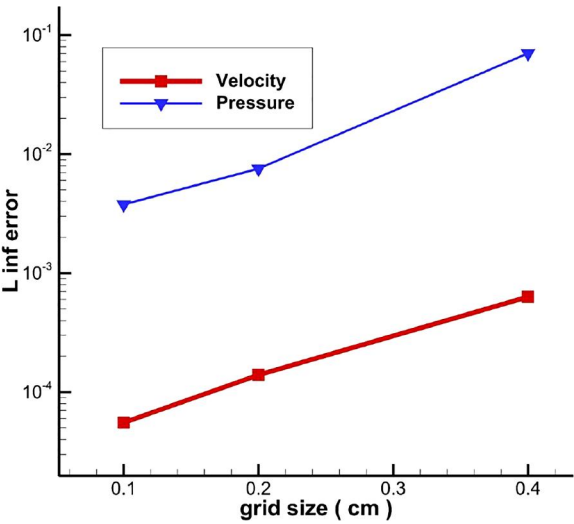


Figure 5. Variation of L_{∞} error with resolution for maximum velocity $|U|_{\max}$ and pressure jump $E(\Delta P_{\max})$.

Table 3. Flow conditions of liquid and gas.

Inlet size (mm)	Ambient pressure P (MPa)	Gas density $\rho_{\text{gas}} \left(\frac{\text{kg}}{\text{m}^3} \right)$	Liquid density $\rho_{\text{liquid}} \left(\frac{\text{kg}}{\text{m}^3} \right)$	Gas viscosity $\mu_{\text{gas}} (\text{Pas})$	Liquid Viscosity $\mu_{\text{liquid}} (\text{Pas})$	Surface tension coefficient $\delta \left(\frac{\text{N}}{\text{m}} \right)$	Liquid velocity $u_l \left(\frac{\text{m}}{\text{s}} \right)$	Gas velocity $u_g \left(\frac{\text{m}}{\text{s}} \right)$
0.1	3	37.25	931.32	2870e-6	18.465e-6	30.0e-3	100	0

and shear stresses, eventually resulting in the stream's breakup into ligaments of liquid, which is known as primary spray breakup. These ligaments continue to experience the same forces and collisions with other ligaments and finally becoming atomized droplets. In general, in fuel injection system, this happens of a very short distance (centimeter). Validating the VOF system for use in spray into a quiescent chamber with the pressure in the chamber at 3 MPa through a small orifice is presented here. The cylindrical chamber has a diameter of 2.31 mm and a height of 9.9 mm. The properties of the fluid materials for both the injected liquid and the chamber gas are given in Table 3.

The first test case is on a coarse mesh, where the domain consists of 93,931 nodes and 90,240 elements that is decomposed into 10 processors as shown in Figure 6. The parallel solution system is described in a paper by Waters and Carrington [19]. The mesh and domain decomposition is shown in Figure 6a. To capture the flow characteristics at the exit of the injector, the mesh size along the injection axis is about one-eleventh of the injector diameter. To save computational time, the mesh size is graded as it expands to the wall as shown in Figure 1b. No-slip wall boundary conditions are imposed at the wall of the cylinder. A constant velocity inlet is specified for the liquid being injected, which is laminar inflow. Free flow boundary conditions (ambient pressure, temperature) along with zero gradient momentum conditions are used at the outlet of cylinder.

Figure 7 shows the velocity and dynamic turbulence viscosity at time = 0.001 s. Since the internal flow is turbulent (Figure 7b) caused by the high Reynolds numbers, turbulence eddies emerging from nozzle may destabilize the interface. We see the velocity difference along the dynamic interface from Figure 7, which is causing Kelvin–Helmholtz instability. Figure 8 shows the liquid volume fraction at time = 0.001 s; here the liquid jet core is smooth over the first five injector diameters and does not show disturbances until around 10 injector diameters which agrees with other published work by Lebas et al. [6] and Shinjo [7].

The density is presented in Figure 9 showing evidence that core's continuity is completely disrupted after about 13 injector diameters. Given this coarse grid, the turbulence is not being well

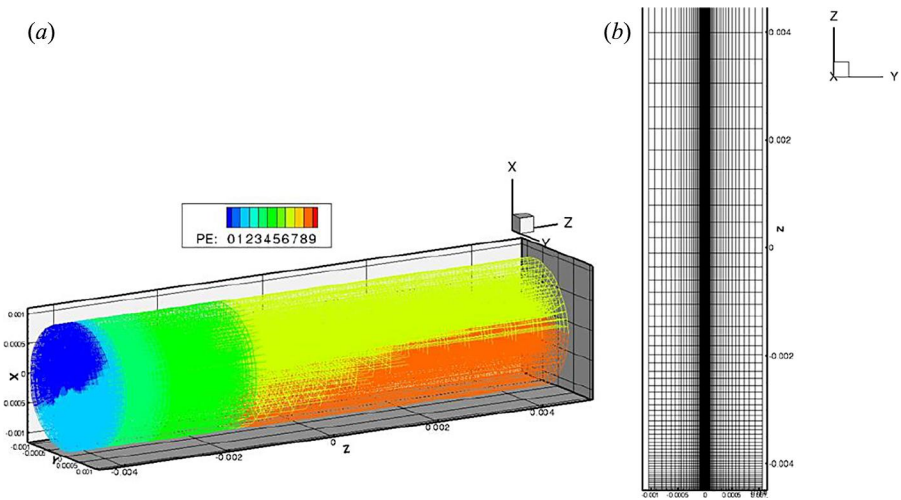


Figure 6. (a) Mesh and decomposition, PE = # of the processors (b) meridional mesh slice at $x = 0$.

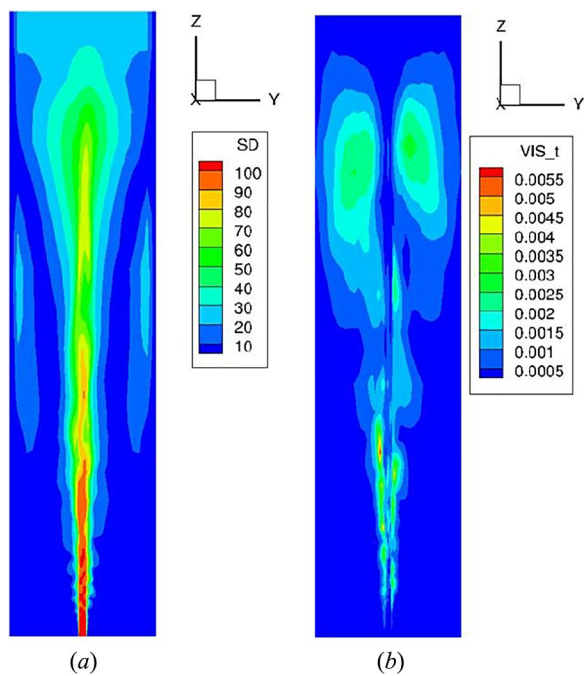


Figure 7. Velocity and turbulent viscosity: SD (velocity magnitude), VIS_t (turbulent viscosity). (a) Velocity slice at $x = 0$, (b) turbulent viscosity slice at $x = 0$.

resolved by the LES algorithm and appears similar to a Reynolds-averaged Navier–Stokes (RANS) solution. This LES method produces solutions very similar to Reynolds-averaged solutions when the mesh coarse (about the type of resolution used with a RANS method). Given the coarse grain resolution provided by the grid, smaller ligaments and even smaller droplets are not being modeled in the simulation. But, the point of the effort shows that the combination of VOF with LES within the KIVA-hpFE FEM projection method has the ability to model the main behavior of the spray’s

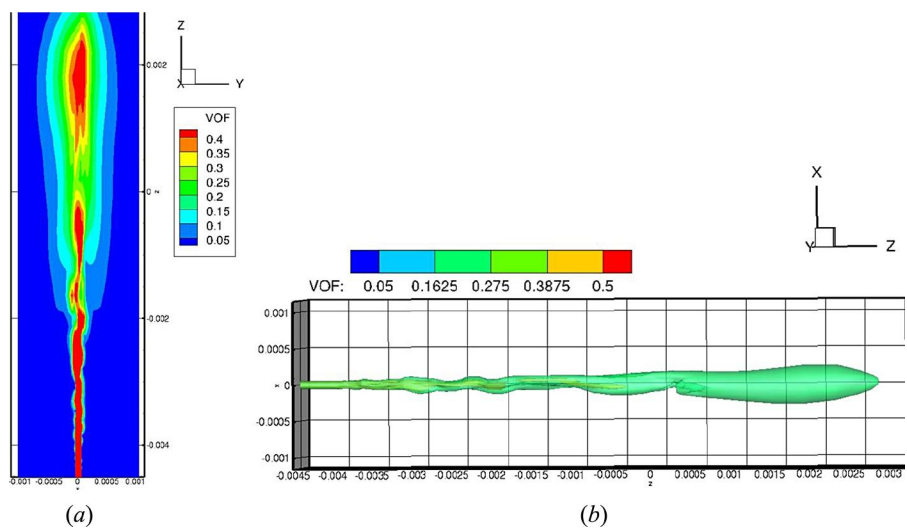


Figure 8. VOF plots (a) $x = 0$ slice, (b) isosurface plot of 3D VOF. *Note:* volume of fluid.

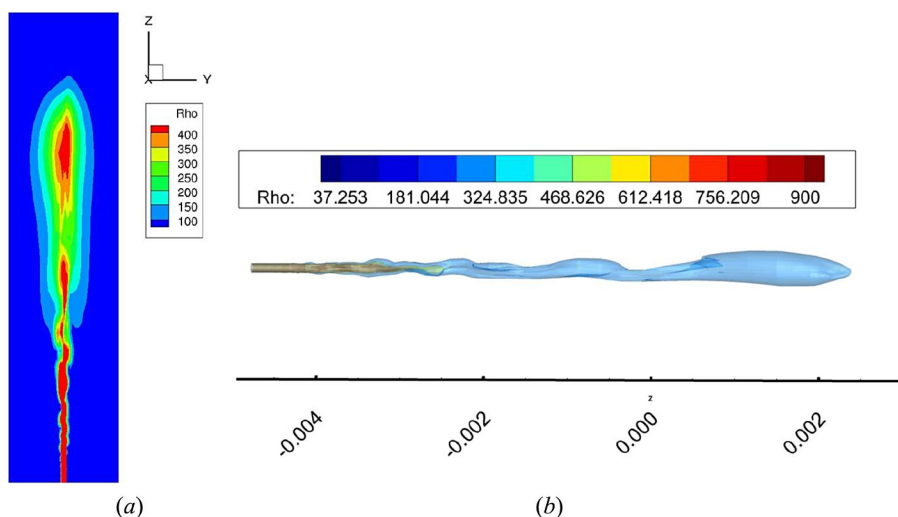


Figure 9. Density plots (ρ) (a) at $x = 0$ slice. (b) isosurface plot of 3D density.

primary breakup and satisfies the prediction from the literature, we have compared the solutions to, notably, Lebas et al. [6] and Shinjo [7].

Studying spatial convergence we use the same setup as before on a finer mesh, however, the mesh is still coarse compared to the resolution required by DNS which is suggested to be $1.5 \mu\text{m}$ grid spacing for a 0.1 mm nozzle (Shinjo [7]). The domain is only more highly resolved near the nozzle, having a $5 \mu\text{m}$ element size for the 0.1 mm nozzle. Figure 10b shows the domain consisting of 499,593 nodes and 491,520 elements that is distributed to 32 processors as shown in Figure 10a. With this finer grid, we are able to model smaller ligaments during liquid core breakup and observe the cause

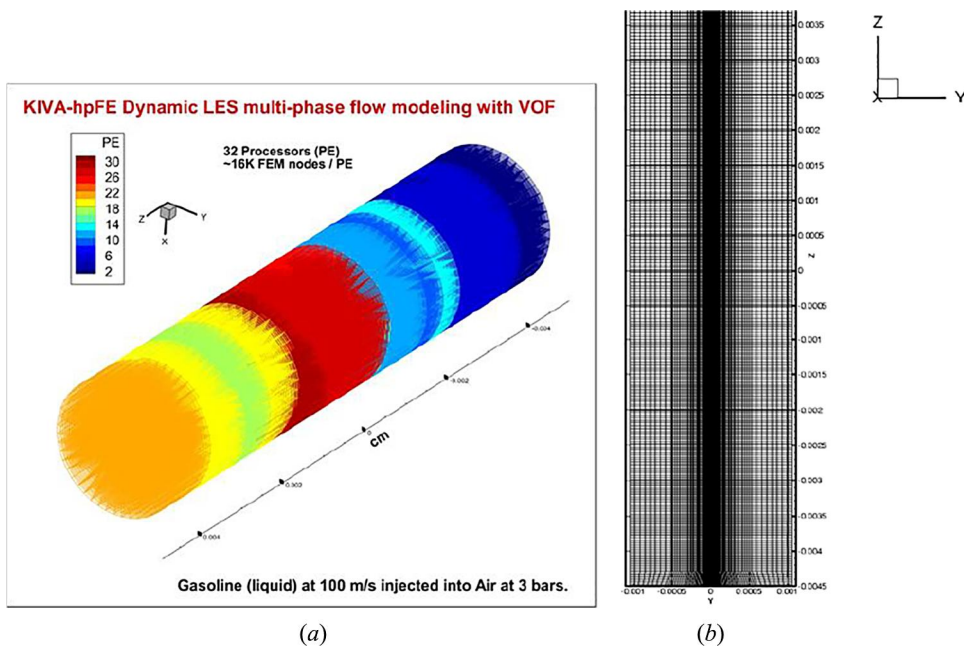


Figure 10. (a) Mesh and decomposition, PE is # of the processors (b) meridional mesh slice at $x = 0$.

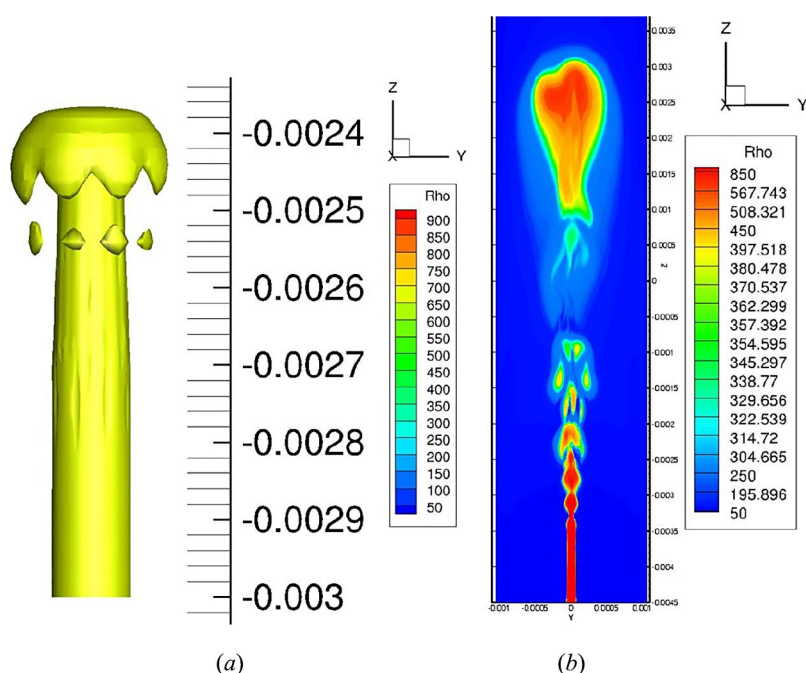


Figure 11. Density time progression showing liquid core breakup. (a) Early in time, (b) later in time.

of the breakup. From this resolution, we obtain a good estimation for initial ligament sizes to be used in the secondary breakup modeling found in the Lagrangian particle transport simulation. Note that since we used a laminar inflow, we only expect the LES to take place once the core starts to become unstable (Xiao et al. [8]).

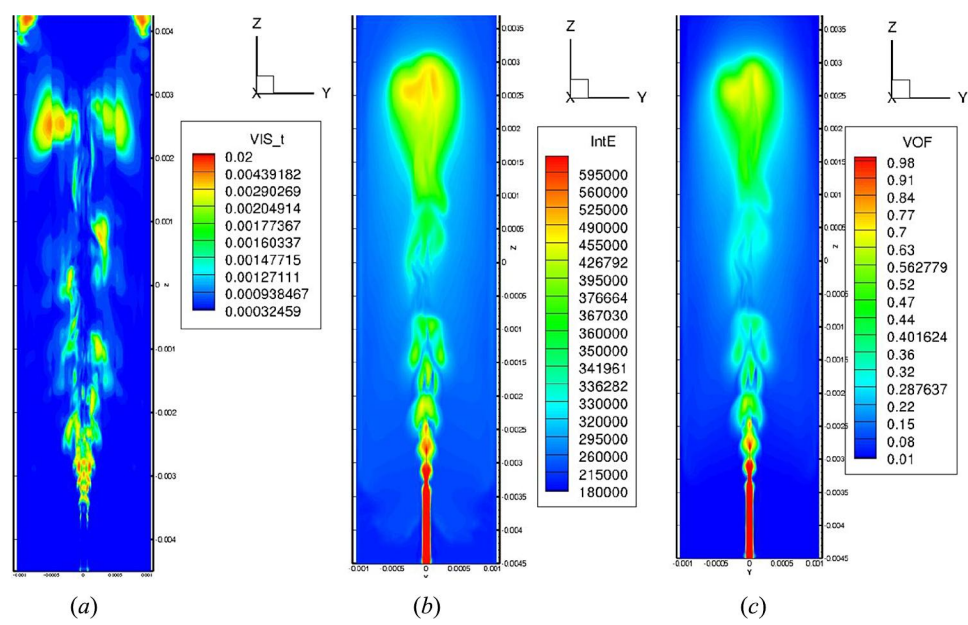


Figure 12. Turbulent viscosity, internal energy, and VOF at late in time. (a) Turbulent viscosity, (b) internal energy, (c) VOF. Note: volume of fluid.

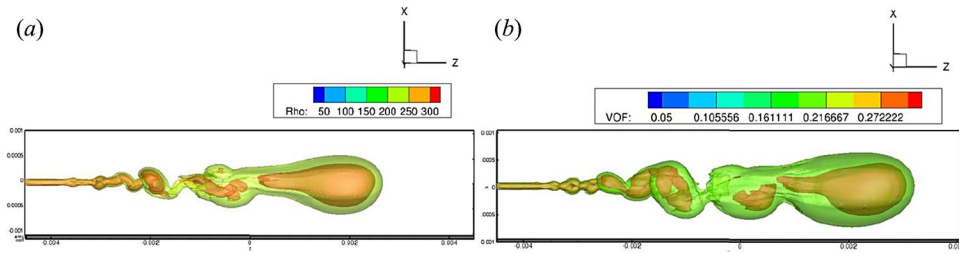


Figure 13. (a) Density and (b) VOF isosurface plots by VOF-LES with laminar inflows. *Note:* volume of fluid.

Figure 11 shows density plots as the solution progresses. In this work, we did not add nozzle disturbances (no turbulence) and the injected liquid velocity is constant. Therefore, the flow instability and primary liquid core breakup is caused by the stress at the interface between the liquid and air. For atomization, since we are not considering the nozzle disturbances, we should expect the formation of mushroom-shaped front and the breakup at early time would be from the tip of the jet as shown in Figure 11a. The mushroom shape of front started to breakup at 5–6 injector diameters. Also the highly disturbed region behind the tip later in time should carry the disturbances through the gas phase and atomization from the core should occur as shown in Figure 11b. The turbulent viscosity is shown in Figure 12a, and it is showing since the inflow is laminar, the flow did not get turbulent until 10 injector diameters and after that, we start to see the small ligament buildup. The internal energy is presented in Figure 12b which is consistent with the VOF development (Figure 12c). We presented the jet structure in Figure 13 and since it is LES with Laminar inflows, it is not as turbulent as turbulent inflow for the liquid. Our result matches the work from Shinjo [7] and Xiao et al. [8] very well. With this resolution study on the same spray problem, we can conclude that higher resolution gives better result for the breakup while the coarse mesh gave the right main flow structure, which is good to predict the statistic of the flow behavior before its secondary breakup. Our VOF-LES enables users to get the right information for the initial input for the next breakup stage with less computational time than DNS.

4. Conclusion

The present study has focused mainly on the implementation of VOF combined with LES in an FEM discretization of the governing equations of fluid flow. Using LES for modeling the turbulence, the primary breakup of liquid is performed on a relatively coarse mesh compared to that needed by more DNS-like simulations. The LES method produces quantifiable and similar results to other published work using DNS. The simulation shown in this paper is for a compressible flow system as opposed to other researchers who mostly perform fully incompressible-flow calculation. The FEM projection scheme enables the solution of one set of momentum equations where an artificial compressibility term is used for the liquid region. The use of Petrov-Galerkin FEM helps improve the accuracy of VOF equation, allowing capture of the interface and removing the need to reconstruct the interface at each time step, while also eliminating the use of mass-flux balancing across the interface between the liquid and gas. Surface tension is a volume force in the FEM weak formulation, it applied at every point that is in the mixed domain where VOF is between 0 and 1. The calculation of interface is easier with the FEM than with FVM and the method does not require smoothing to calculate the gradient of the color function. We demonstrated the system for injection and spray modeling. The way of starting to validate the methods and their implementation comparisons is provided with published results; showing good agreement. Differences between DNS results and the demonstrated LES and liquid core breakup is a result of differences in the grid resolution. The method as implemented, the first time VOF and LES are shown in an FEM-type weak form for modeling injection and liquid stream's core

primary breakup. Particularly considering, both compressible gas and incompressible liquid are solved together and simultaneously.

Funding

The DOE's Office of Energy Efficiency and Renewable Energy (EERE) Advanced Combustion Program (Gurpreet Singh and Leo Breton) is supporting this effort. Los Alamos National Laboratory, an affirmative action/equal opportunity employer, is operated by the Los Alamos National Security, LLC for the National Nuclear Security Administration of the U.S. Department of Energy (DOE) under contract DE-AC52-06NA25396. Los Alamos National Laboratory strongly supports academic freedom and a researcher's right to publish; as an institution, however, the Laboratory does not endorse the viewpoint of a publication or guarantee its technical correctness.

References

- [1] A. A. Amsden, P. J. O'Rourke, and T. D. Butler, "KIVA-II: A computer program for chemically reactive flows with sprays," Los Alamos National Laboratory Scientific Report, LA-11560-MS, 1989.
- [2] D. J. Torres and M. F. Trujillo, "KIVA-4: An unstructured ALE code for compressible gas flow and sprays," *J. Comp. Phys.*, vol. 219, pp. 943–975, 2006.
- [3] J. K. Dukowicz, "A particle-fluid numerical model for liquid sprays," *J. Comp. Phys.*, vol. 35, no. 2, pp. 229–253, 1980.
- [4] D. B. Carrington, "A fractional step hp-adaptive finite element method for turbulent reactive flow," Los Alamos National Laboratory Report, LA-UR 11-00466, 2011.
- [5] D. J. Torres, P. J. O'Rourke, and A. A. Amsden, "Efficient multi-component fuel algorithm," *Combust. Theor. Model.*, vol. 7, pp. 67–86, 2003.
- [6] R. Lebas, T. Menard, P. A. Beau, A. Berlemont, and F. X. Demoulin, "Numerical simulation of primary break-up and atomization: DNS and modelling study," *Int. J. Multiphase Flow*, vol. 35, no. 3, pp. 247–260, 2009.
- [7] J. Shinjo and A. Umemura, "Simulation of liquid jet primary breakup: Dynamics of ligament and droplet formation," *Int. J. Multiphase Flow*, vol. 36, no. 7, pp. 513–532, 2010.
- [8] F. Xiao, M. Dianat, and J. J. McGuirk, "LES of turbulent liquid jet primary breakup in turbulent coaxial air flow," *Int. J. Multiphase Flow*, vol. 60, pp. 103–118, 2014.
- [9] C. W. Hirt and B. D. Nichols, "Volume of fluid (VOF) method for the dynamics of free boundaries," *J. Comput. Phys.*, vol. 39, no. 1, pp. 201–225, 1981.
- [10] M. M. Francois et al., "A balanced- force algorithm for continuous and sharp interfacial surface tension models within a volume tracking framework," *J. Comput. Phys.*, vol. 213, pp. 141–176, 2006.
- [11] M. S. Kim and W. I. Lee, "A new VOF-based numerical scheme for the simulation of fluid flow with free surface. Part I: New free surface-tracking algorithm and its verification," *Int. J. Numer. Method Fluids*, vol. 42, pp. 765–790, 2003.
- [12] T. Ménard, S. Tanguy, and A. Berlemont, "Coupling level set/vof/ghost fluid methods: Validation and application to 3d simulation of the primary break-up of a liquid jet," *Int. J. Multiphase Flow*, vol. 33, no. 5, pp. 510–524, 2007.
- [13] G. Vaudora, T. Ménarda, W. Aniszewskic, M. Doringb, and A. Berlemont, "A consistent mass and momentum flux computation method for two phase flows. Application to atomization process," *Computers Fluids*, vol. 152, pp. 204–216, 2017.
- [14] C. C. Yu and J. C. Heinrich, "Petrov-Galerkin methods for the time-dependent convective transport equation," *Int. J. Numer. Method Eng.*, vol. 23, pp. 883–901, 1986.
- [15] J. Waters, D. B. Carrington, and D. W. Pepper, "An adaptive finite element method with dynamic LES for turbulent reactive flows," *Comput. Therm. Sci.*, vol. 8, pp. 57–71, 2016.
- [16] M. Herrmann, "Detailed numerical simulations of the primary atomization of a turbulent liquid jet in crossflow," *ASME J. Eng. Gas Turbines Power*, vol. 132, p. 061506, 2010.
- [17] O. C. Zienkiewicz, R. L. Taylor, P. Nithiarasu, and Ebooks Corporation 2013, *Finite Element Method for Fluid Dynamics*, 7th ed. Burlington: Elsevier Science, 2013.
- [18] D. B. Carrington, X. Wang, and D. W. Pepper, "A predictor-corrector split projection method for turbulent reactive flow," *Comput. Therm. Sci.*, vol. 5, no. 4, pp. 333–353, 2013.
- [19] J. Waters and D. B. Carrington, "A parallel Large Eddy Simulation in a finite element projection method for all flow regimes," *Numer. Heat Transfer, Part A*, vol. 70, no. 2, pp. 117–131, 2016.
- [20] J. U. Brackbill, D. B. Kothe, and C. Zemach, "A continuum method for modelling surface tension," *J. Comput. Phys.*, vol. 100, pp. 335–354, 1992.
- [21] A. Beliveau, A. Fortin, and Y. Demay, "A two-dimensional numerical method for the deformation of drops with surface tension," *Int. J. Comp. Fluid Dynamics*, vol. 10, pp. 225–240, 1997.
- [22] M. Herrmann, "A balanced force refined level set grid method for two-phase flows on unstructured flow solver grids," *J. Comput. Phys.*, vol. 227, pp. 2674–2706, 2008.

Appendix for FEARCE

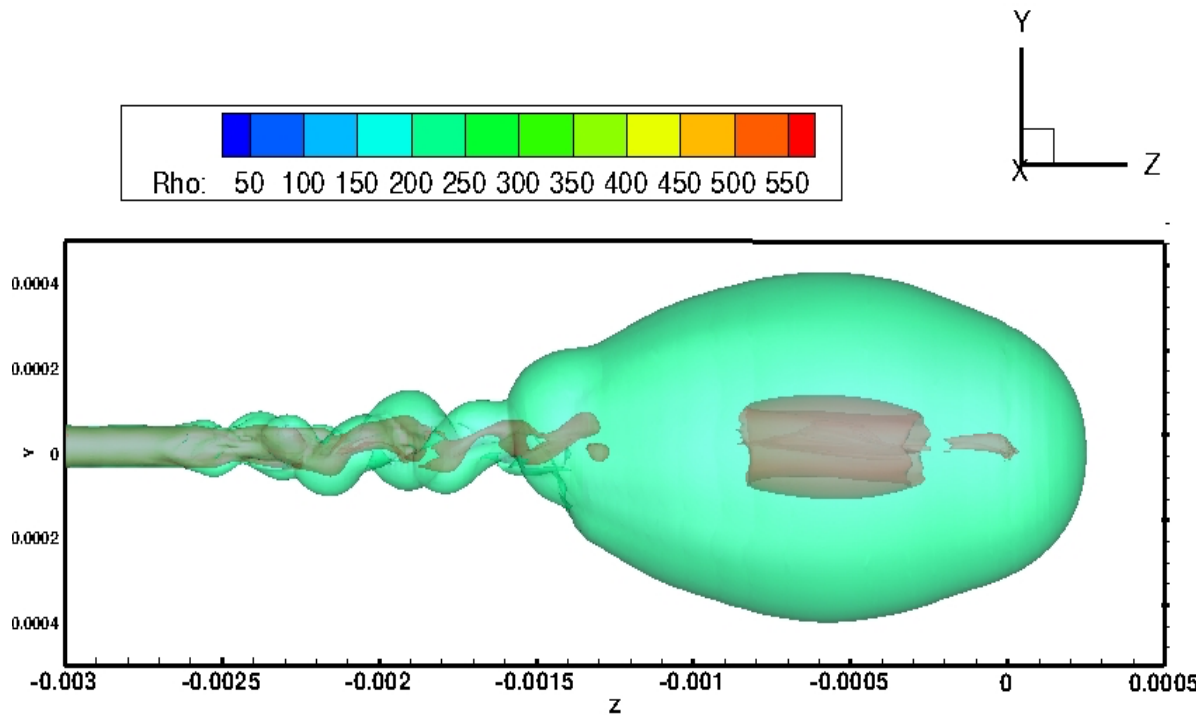


Figure 1. This figure shows liquid jet breakup in a gas using FEARCE's Volume-of-Fluids feature. Note the centerline slide of the three-dimensional solution, which shows the very high density liquid jet breaking up into ligaments.

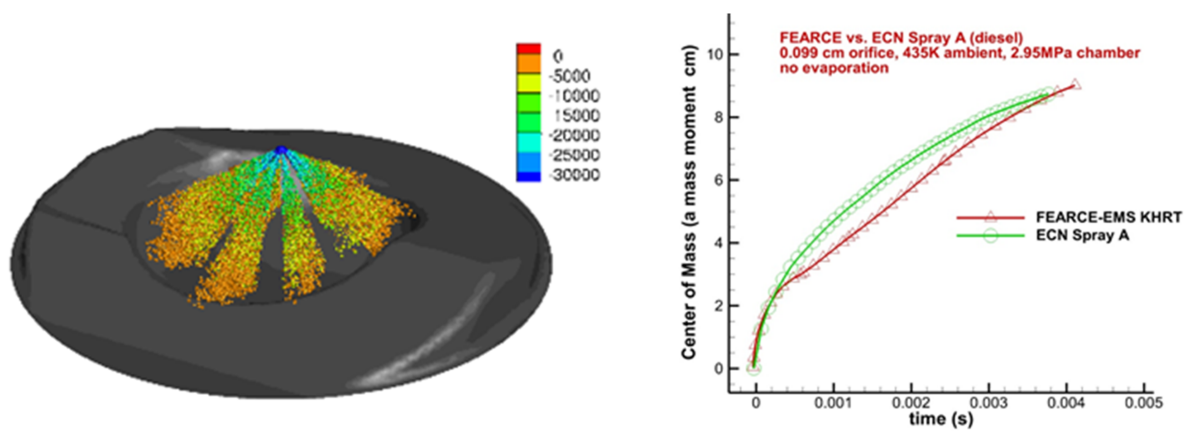


Figure 2. This figure shows KH-RT injected liquid jet breakup and spray atomization produced in FEARCE.

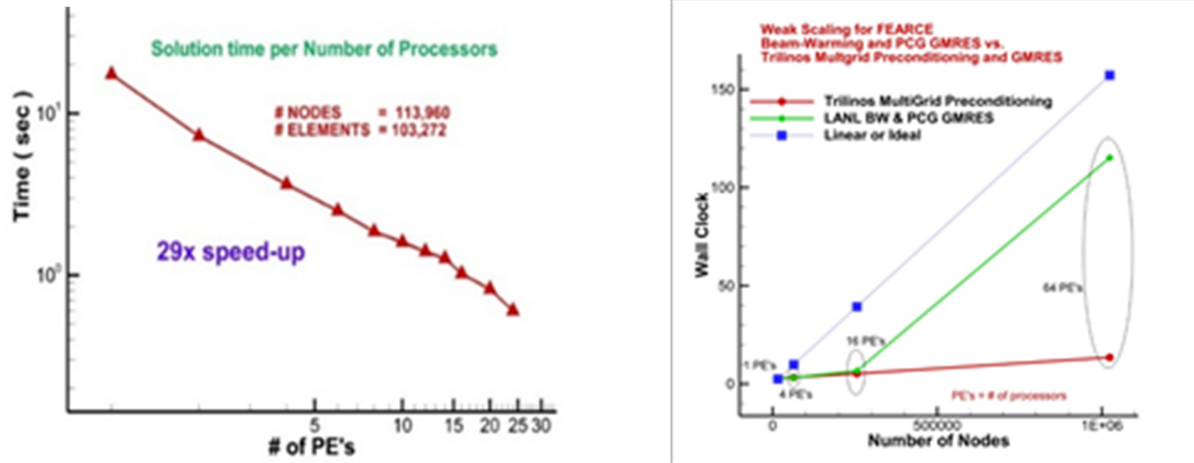


Figure 3. These graphs show FEARCE scaling (1) strong (same size with more processors) showing superlinearity of algorithm and (b) weak (size and processors doubling) showing superlinear algorithm, no preconditioning, and a very flat response to double with multigrid preconditioning.

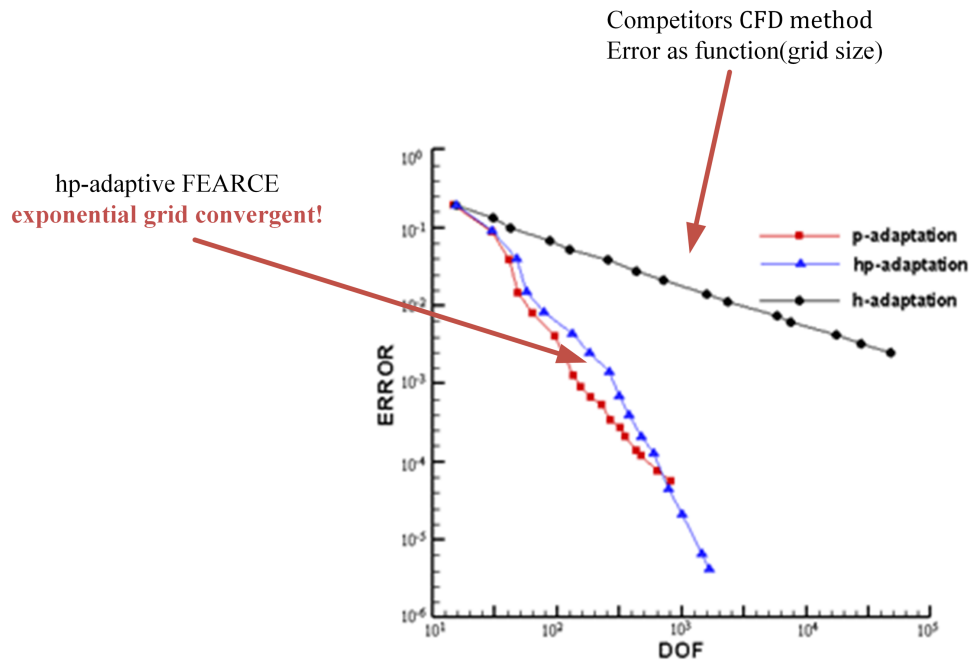


Figure 4. This graph shows FEARCE's hp-adaptive convergence rate versus that of the competition, which uses only a grid-resolution enhancement. Although the hp-adaptive method is on the same curve as p-adaptive (higher order) it is much less expensive than higher order used alone.